MIKROSZÁMÍTÓGÉP MAGAZIN

A NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG LAPJA

Ára: 30 Ft

1986 február

C Számítástechnikai Áruház

Budapest XIII., Balzac u. 35.

Köszöntjük Önöket, kedves Olvasók!

Bemutatkozásunkat egy nagy visszhangot kiváltó mikroszámítógépes bemutató előzte meg. Ekkor határoztuk el, hogy ismertetjük termékünket a Mikroszámítógép Magazinban, és az olvasók bizalmába ajánljuk a

RAAB 84

számítógépcsaládot.

Fejlesztője a Controll Elektronikai Társaság. A számítógépcsalád kifejezetten a professzionális ügyviteli feladatok megoldására készült. Főbb alkotóelemei a következők.

RAAB 84/A alapkonfiguráció

Z80A processzorral, 64 kbájt RAM és 32 kbájt EPROM memóriával, max. 4 db 5 1/4 ″(1 Mbájt) vagy 8″ floppyegység fogadására alkalmas kivitelben. Alapkiépítésben 2 db mini floppyegységet tartalmaz. A standard számítástechnikai klaviatúra magyar ékezetes betűkkel, külön funkció- és numerikus billentyűzettel van ellátva. A tasztatúráról választható az angol vagy a magyar ábécé. A képernyőn 80 × 24 karakter vagy 160 × 72 kvázigrafikus képpont jeleníthető meg színesen vagy fekete-fehéren.
A beépített nyomtatóinterfész tetszőleges típusú printer illesztésére szolgál, soros és párhuzamos csatlakozási lehetőséggel.
Operációs rendszere a CP/M 2.2 kompatibilis MC/DOS COBASIC BASIC interpreter.

RAAB 84/W Winchester-lemezes konfiguráció

A RAAB/A alapgép Winchester-lemezes háttértárolóval kiegészített változata. Max. 2 db 10, 19, 27, 40 Mbájt kapacitású lemez fogadására alkalmas kivitel. Alapkiépítésében 1 db 27 Mbájtos Winchester- és 2 db 1 Mbájtos mini floppylemezt tartalmaz. A lemezoperációs rendszer lineárisan kezeli a Winchester-lemezes háttértárolót, annak kapacitásától függetlenül.

RAAB 84/T
intelligens terminál a lokális hálózatban

Háttértároló nélküli alapgép, a számítógép-hálózat önálló munkahelye. A felhasználói programok a terminálegységben futnak, és az operációs rendszeren keresztül érik el más gépek háttértárolóit.
A számítógépegységek kommunikációs hálózaton keresztül történő összekapcsolása a következő előnyökkel jár.
— Bármely egység elérheti egy másik egység háttértárolóját. A terminálok, melyek a kommunikációs hálózaton keresztül egy másik egység háttértárolóját használják, teljes értékű alapgépfunkciót töltenek be.
— A hálózati operációs rendszer intelligens módon támogatja a programok közötti nagy sebességű kommunikációt.
— A kívánt konfiguráció lépésenként, fokozatosan bővítve, tetszés szerint alakítható ki.

Referenciahelyeink

Győri Szeszipari Vállalat
Győr-Sopron Megyei Tanácsi Építőipari Vállalat, Győr
SZÜV Dialóg Leányvállalat, Győr
Bősárkányi MGTSZ
Somogy Megyei Építőipari Vállalat, Kaposvár
TSZ Társulás, Szekszárd

A RAAB 84 számítógépcsalád sikeréhez hozzájárul az előnyös ár, a gyors szállítás és a cseregépes szerviz.

Szoftverszolgáltatásaink

— Programozási nyelvek: BASIC, PASCAL, ASSEMBLER, PL/I, FORTRAN
— Szövegszerkesztők
— Anyag- és készletgazdálkodási rendszerprogramok
— Egyedi elképzeléseket megvalósító szoftver részleg
Megkeresésükre a gyümölcsöző partnerkapcsolat reményében örömmel adunk felvilágosítást.

Új Kalász MGTSZ
Számítástechnikai ágazat
9023 Győr, Bacsó Béla u. 65.
Telefon: 15-858
Telex: 24-466

MIKROSZÁMÍTÓGÉP MAGAZIN

4. ÉVFOLYAM
1. SZÁM
1986. FEBRUÁR

A NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)
Broczkó Péter
(hírek)
Kovács Győző
(levelezés)
Lindner László
(sakkprogramozás)
Petróczy Judit
(könyvek)
Pogány Csaba
(alkalmazástechnika, tanfolyam)
Simonyi Endre
(klub)
Takácsy Ildikó
(favágás)
Vadkerti János
(μprogramok)
Varga András
(iskola — számítógép)
Votisky Zsuzsa
(játékprogramok)

A szerkesztőség tagjai:

Kardos Zsuzsa
Nacsa Sándor

Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő u. 68.
Telefon: 154-250

Kiadja a Delta Szaklapkiadó és Műszaki Szolgáltató Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest VII., Garay u. 5.
Telefon: 415-583, 215-440

Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban,
a kézbesítőknél,
a Posta hírlapüzleteiben
és a Hírlapelőfizetési
és Lapellátási Irodánál
(Budapest V., József nádor tér 1.
Postacím: 1900 Budapest)
közvetlenül
vagy postautalványon,
valamint átutalással
a HELIR 215—96162
pénzforgalmi jelzőszámra.
Megjelenik havonta
Példányonkénti ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft

85-3587, Szikra Lapnyomda
Budapest
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25629
ISSN 0236-6088

**Címképünk:
a NOVOTRADE RT
2C Számítástechnikai
Áruháza**

Tartalom

Évente tizenkétszer — új köntösben 2
A minőségügy közügy 5
Akik csatát veszíthetnek, de háborút nem 19
Mikroinformatikai forradalom Lengyelországban 20
A nyomtatástechnika újabb eredményei 24
Közlekedő- és számolóedények 40

ISKOLA — SZÁMÍTÓGÉP

Az általános iskolai kísérleti számítástechnikai program 3
Hasznos gépi rutin 4
HT—BASIC DATA — újból 6
Futás közben változó programok 8

TANFOLYAM

Alapozás XIV. 10

PROGRAMOZÁSTECHNIKA

Kis gépek — nagy programok 14
Strukturált programtervezés 16
A LOGO új változata a SmartLOGO 22

PIAC

2C Számítástechnikai Áruház 26
Új módszerek a kereskedelemben 29

TERMÉKISMERTETŐ

A Commodore 16 27

μPROGRAMOK

Karakterszerkesztő EPSON nyomtatóhoz 30
Megszakítás II. 32
Módosítások Commodore gépeken 33
Príma tippek 33

μKLUB

A hiba nem az Ön készülékében van 34
Klubok és iskolák 36

JÁTÉKPROGRAMOK 37

AZ OLVASÓ ÍRJA 42

FAVÁGÁS

Gyakorlatok késleltetőkkel 44

SAKKPROGRAMOZÁS

Bitek és figurák 46

HÍREK, ÉRDEKESSÉGEK 47

Évente tizenkétszer — új köntösben

„'S ha szerencsétlen eleinket a' még tisztára ki nem fejlett korok előítéletei, természetes vadságokra, belső irígykedés 's külső fondorkodások örök meghasonlásokra ingerelték, 's ők sokszor, úgy szólván, a' hon megrontására esküdtek össze: kössünk mi újabbak, műveltebbek már valahára egy szebb frigyet, melly alatt nem egyesek haszna 's előmenetele lappang, hanem a' közjó melegítő sugára fényledez."

(SZÉCHENYI ISTVÁN: STÁDIUM)

Furcsa arra gondolnom, hogy a µM negyedik évfolyamát kezdjük, hiszen még élénken élnek emlékezetemben az 1983-as szám kiadása körüli polémiák, a lap alapításával és az engedélyeztetéssel kapcsolatos gondok, optimizmusunkat megrendíteni akaró pesszimista jóslatok.

Lassan három éve történt mindez.

Voltaképpen elégedettek lehetnénk, hiszen az elmúlt években kialakult a µM tábora, kapcsolatunk az olvasókkal folyamatosan erősödik. Azt hiszem, valamennyi lap számára a legfontosabb hírforrás az olvasók levelei. Mi — örömmel mondhatom — változatlanul igen sok levelet kapunk, dicséretet is, kritikát is, egyre több írást, véleményt, programot és javaslatot — egyszóval élvezzük az olvasók érdeklődését, amire minden szerkesztőségnek — így nekünk is — nagyon nagy szükségünk van.

Szinte az első szám szerkesztése óta állandó problémánk, hogy a lap rendszertelenül jelenik meg, néha hónapokig egyetlen szám sem, aztán az év vége felé havonta egy-egy. Egy korábbi szerkesztőségi cikkben már egyszer elmondtam, hogyan készül a µM, azt is, hogy a szerkesztőség majdnem hathónapos előretartással dolgozik, ennek ellenére képtelenek voltunk elérni, hogy a lap a tervezett időpontban megjelenjék. Szerződéseink voltak és vannak, az abban közösen elfogadott határidőket mégsem tudták partnereink tartani. Lehet, hogy furcsán hangzik, de a havi megjelenéstől reméljük, hogy ez a rendszertelenség megváltozik, képesek leszünk arra, hogy a lap a hónap előre meghatározott napján, késés nélkül megjelenjék. Havilapnál ugyanis nincsenek tartalék idők, ha egyszer a lap késik, akkor a késést gyakorlatilag nem lehet behozni. Így már nemcsak a szerkesztőségnek lesz érdeke, hogy a lap elkészüljön, de a nyomdának is, ami garancia lehet a pontosságra.

1986-tól az évi tizenkétszeri megjelenéssel olvasóink régi kívánságának is eleget teszünk; leveleiknek kb. egynegyedében követelik a többszöri megjelenést.

Nem tagadom, vonzottak bennünket a sűrűbb megjelenésből származó előnyök is. Ezek közül az egyik, hogy az átfutási idő automatikus csökkenésével lehetőségünk lesz olyan cikkek, riportok, beszámolók közlésére is, amelyekre a hosszú átfutási idő miatt eddig még csak nem is gondolhattunk. (Szabad legyen itt az elmúlt évi linzi írásomra hivatkozni, amelyben — naivan — az Ausztriában érvényes számítógépárakat is felsoroltam. Mondanom sem kell, hogy ezek az árak a lap megjelenésekor — négy hónappal a megírás után — már messze nem voltak érvényesek.) Eddig nem volt értelme például, hogy más újságokban megjelent írásokra válaszoljunk, vagy azokkal vitatkozzunk, hiszen néhány hónap után már senki sem emlékezett vissza azokra az eredeti írásokra, amelyekkel kapcsolatban a véleményünket kifejtettük. Megpróbáltuk azt is, hogy a hazai számítástechnika egy-egy friss eseményéről riportot készítsünk, de beláttuk, hogy erőfeszítésünk hiábavaló, a friss riport néhány hónap alatt elveszti aktualitását, érdektelenné válik.

Talán nevetségesen hangzik, de nagyon sok problémám van például az „Olvasó írja" rovattal is. Nemegyszer kapunk szatirikus levelet, amelyben a levélíró megköszöni, hogy közöltem a levelét, amelyben például olyan valakit keres, akinek hasonló számítógépe van, mint neki. Elmondja, hogy mire a levele megjelent, addigra a számítógépét már eladta, sőt azt is, amit utána vett.

Ez volt az oka — és ezt olvasóink bizonyára észrevették —, hogy a µM szerkesztését az első számok után alaposan meg kellett változtatnunk. Elhagytuk az ún. friss riportokat, a nagyon aktuális mondanivalójú cikkeket, csak lassan „avuló" munkák maradtak meg, elsősorban a programok, oktatási anyagok, hosszú távra értékelő, esetleg polemizáló írások.

Pedig nemcsak az olvasók leveleiből tudjuk, hanem magunk is érezzük, hogy friss és gyors tájékoztatás nélkül a µM fejlődése előbb vagy utóbb megáll, olvasóink egy részének érdeklődését elveszítjük.

Ezért teremtettük meg az átfutási idő csökkentésének a technikai feltételeit, hogy az egyes műveleteket (szedés, korrektúra, nyomtatás stb.) az eddiginél sokkal rövidebb idő alatt tudjuk végrehajtani. Úgy tűnik, az idővel való gazdálkodás sikerrel járt, és így 1986-ban a lap havonta és pontosan ugyanazon a napon fog megjelenni.

Az elmondottakon kívül más tartalmi változást nem tervezünk. Megtartjuk az eddigi rovatszerkezetet, ha sokakat érdeklő új témát találunk, akkor habozás nélkül új rovatot indítunk. Egy ilyen újdonságon már most gondolkozunk, ez pedig a robotika. Úgy látjuk, hogy egyre többen foglalkoznak a robotokkal, nemcsak az iskolákban, az oktatómunka keretében, de amatőrök is szívesen használják a számítógépeket robotok vezérlésére. Ha a témának gazdát találunk, szívesen közlünk a robottechnikáról is dolgozatokat.

Talán emlékeznek jelmondatunkra: *együtt az olvasókkal!* Havilappá válásunk az olvasók tevékeny közreműködése nélkül egész biztosan nem lenne sikeres. Nemcsak arra gondolok, hogy a jövőben legalább kétszer annyi írásra lesz szükségünk, hanem arra is, hogy egy havilap csak előfizetői rendszerben tud megélni, egyedi árusításból valószínűleg nem. Ezért akciót indítunk előfizetők toborzására, mert nagyon szeretnénk, ha a µM-t majdnem teljes egészében ilyen úton terjesztenék, és a példányok csak egy kisebb hányadát a hírlapárudák segítségével.

Sokszor ígértük, de még mindig nem tudtuk megoldani a lap külföldi terjesztését. Az érdeklődés elsősorban a szomszédos országok magyar ajkú lakossága részéről jelentkezik. Én azt remélem, ha a µM-ban sorozatban közlünk a hazai számítástechnika eseményeiről friss riportokat, aktuális mondanivalójú írásokat, akkor ezzel fel tudjuk kelteni távolabbra szakadt hazánkfiai érdeklődését is. Ezért kell a külföldi terjesztés kérdését nagyon gyorsan megoldanunk.

Amikor először beszéltünk a gyakoribb megjelenésről, akkor az egyik barátom azt mondta: „Miért nem vagytok elégedettek azzal, hogy van egy sikeres lapotok, amelyhez megvannak a külső és belső munkatársak, kaptok elegendő cikket, így kényelmesen meg tudjátok szerkeszteni azt az évi hat számot. Hát érdemes hajszolni magatokat, állandó időzavarral küszködni, és a végén ráfizetni?!"

Ugye érdemes?

KOVÁCS GYŐZŐ

Az általános iskolai kísérleti számítástechnikai programról

Rovatunkban az iskolai munkához nyújtott közvetlen segítségen, illetve egy-egy iskola tapasztalatainak ismertetésén túlmenően időközönként az országos iskolaszámítógép-programmal kapcsolatos információkat is szeretnénk közvetíteni. Ezért kértük meg a Tudományszervezési és Informatikai Intézet és az Országos Pedagógiai Intézet felelős munkatársait, ismertessék azt a közös erőfeszítéssel végzett előkészítő munkát, amelynek célja a programnak az általános iskolákra való kiterjesztése. Várjuk olvasóink további híreit az általános iskolákban megindult munkáról.

Hazánkban az 1982/83. tanévben indult meg az iskolaszámítógép-program keretében a középfokú oktatási intézmények személyi számítógépekkel történő ellátása, a számítástechnikai alap- és alkalmazási ismeretek oktatása ezekben az iskolákban.

Az 1985/86. tanévben amellett, hogy folytatódik a középiskolai számítástechnikai program, megkezdődött az általános iskolai kísérleti számítástechnikai program is. Az ezt előkészítő pedagógiai kutatások már a VI. ötéves tervben megindultak, de az eredmények gyakorlati kipróbálásához előbb meg kellett teremteni a számítástechnikai oktatás eszközhátterét. Megkezdődött a számítógépek telepítése az általános iskolákba.

A magyar ékezetes betűkkel ellátott gépeket hosszadalmas, bonyolult folyamat eredményeképpen sikerült beszerezni, nagyon előnyös áron. Utóbbiban része volt az importáló Skála-Coopnak, amely ily módon elő kívánta segíteni a számítástechnika oktatásának minél szélesebb körű terjesztését.

A számítógépek egy részét a Tudományszervezési és Informatikai Intézettől (TII) kapták az iskolák, másik részét a megyei és városi tanácsok vásárolták, továbbá az iskolák saját költségükre is vettek számítógépeket. Így központi keretből az Országos Pedagógiai Intézet (OPI) által irányított számítástechnikai kísérletben részt vevő iskolák egyenként 8-10, a tanárképző és tanítóképző főiskolák 4-4, ezek gyakorló iskolái 4-4, a megyei Pedagógus Továbbképző Intézetek 2-2 C16 típusú számítógépet kaptak. Ezenkívül több olyan általános iskola kapott számítógépet, ahol biztosítva látszott a gépek megfelelő használata.

A következő években fokozatosan bővülni fog a számítógéppel ellátott iskolák köre, és így egyre több oktatási intézmény kapcsolódhat be a kísérleti programba. Addig azonban, amíg az ország közel 3600 általános iskolájának mindegyikében működni fog egy 10-15 számítógéppel felszerelt laboratórium, még elég hosszú időnek kell eltelnie. Ezalatt meg kell teremteni az eszközellátáson túl a számítástechnikai ismeretek oktatásának többi feltételét is.

Elsősorban meg kell oldani az általános iskolai tanárok felkészítését. A gépek szétosztásával egy időben az érdekelt iskolák egy-két tanára rövid továbbképzésben részesült. Sok általános iskolai tanár már az 1984/85. tanévben részt vett a megyei Továbbképző Pedagógiai Intézet által szervezett tanfolyamokon. Ezeket a tanfolyamokat folytatva és a nagy érdeklődésre való tekintettel számukat növelve néhány év alatt elérhető, hogy az általános iskolai tanárok jelentős része rendelkezzen valamilyen számítástechnikai ismerettel. Az általános iskolai tanárjelöltek számítástechnikai felkészítése ezután már a főiskolákon megtörténik.

Fontos feladat olyan módszertan és tananyag kidolgozása, amely egyrészt a számítástechnikai ismeretek oktatását a közismereti tantárgyak keretén belül oldja meg, másrészt a számítástechnikai módszerek alkalmazását beépíti a különböző tantárgyakba, az adott területen megismerteti a számítástechnika alkalmazási lehetőségeit, és elősegíti a tantárgyi ismeretanyag megértését a személyi számítógép oktatógépként vagy szimulálóeszközként történő alkalmazásával.

Ennek érdekében az OPI-ban 1985-ben elkezdődtek az általános iskolákra vonatkozó kísérletek. A munka fő célja annak feltárása, hogy miként lehet a személyi számítógép felhasználási területeit a tantervi anyaggal és a követelményekkel összhangba hozni úgy, hogy a tanítási-tanulási folyamatban minőségjavulást, időmegtakarítást érjünk el, és informatikai gondolkodásmódot alakítsunk ki. A kísérletek eleinte csak gyenge szállal fűződtek a C16-os géphez. Ugyanis akkor még csak egy kísérletező iskolánkban volt egyetlen C16-os gép. Ennek segítségével több mint 300 példa készült el egy példatár anyagaként. A példákat a pedagógiai tapasztalatokról szóló leírások egészítik ki, ezáltal alkalmasak mind szakköri, mind tantárgyi felhasználásra. A példatár még ebben a tanévben megjelenik.

További számítástechnikai-pedagógiai szakanyagok elsősorban a középiskolai programhoz készültek el, többek között a fizikához (ezen belül külön az elektronikához és mechanikához), kémiához, történelemhez, matematikához, valamint az anyanyelvhez. Ezeknek és a középiskolák számára készült, a TII és a Deltasoft által közösen tejesztett programoknak az általános iskolai sajátosságokhoz adaptálása folyamatosan történik.

Ezzel párhuzamosan kiterjesztjük a TII korábbi programpályázatát részben további számítógéptípusokra, részben a középfokúról az alsó fokú oktatásra.

A következő, jól körülhatárolható területeken, amelyeknek közös jellemzője, hogy nem kötődnek konkrét géptípushoz, további kutatás folyik:

— hogyan lehet a személyi számítógépet az iskolai kísérletek kiértékelésére, vezérlésére, szabályozására felhasználni;

— a videotechnika, a perifériák, az iskolai robotok, a hagyományos audiovizuális eszközök rendszerbe szervezése a személyi számítógép segítségével;

— különböző géptípusok telepítése egy nagyobb (professzionális vagy félprofesszionális) gép „körül";

— olyan tanulói kísérletező készlet és olyan, a tanulók által elkészíthető munkadarabok kifejlesztése, összeállítása, amelyek a személyi számítógép működésének megértését és/vagy az informatikai gondolkodásmód kialakulását segítik elő;

— a tantervek, tankönyvek, taneszközök együttes elemzése részben abból a célból,

hogy elősegítsük a tanulók, a tanárok készségének kialakulását a felhasználást illetően, részben azért, hogy képet kapjunk arról, hogy az informatikai kultúra hazai honosítása hogyan hat a tananyag tartalmára, elrendezésére és a módszerekre; milyen új elemek jelenhetnek meg a taneszközök fejlesztésében, a tankönyvek és más nyomtatott segédletek műfaji kritériumaiban;

— szoftverek tesztelése az iskolákban;
— ergonómia, szociológia.

Az általános iskolákban az 1985/86-os tanévben megindult program minden bizonnyal serkenteni fogja a fejlesztést, rangsorolja az egyes kutatási témákat fontosság szempontjából, és új színeket, lehetőségeket és korlátokat hoz a felszínre. Aligha hihetjük azonban, hogy ebben a témában ugyanazok az eljárások, mintázatok hatásosak lehetnek, amelyeket ma a központi és a helyi rendszerek között találunk (például központilag kiadott tematika vagy tanterv és ehhez segédlet vagy tankönyv). Ha valami éltetni fogja ezt az ügyet, az a sokszínűség, sokféleség és önállóság az iskolákban, a bőséges országos, központi kínálat.

A kísérleti időszakban (néhány évig) a számítástechnikai oktatás az általános iskolákban elsősorban a szakkörökön és fakultációban történik. Az oktatási programok egyre növekvő számával a már számítógéppel rendelkező iskoláknak meglesz azonban a lehetőségük arra, hogy a különböző tantárgyak keretén belül bemutassák a számítástechnika alkalmazásának módszereit, és a számítógép segédeszközként való alkalmazásával megkönnyítsék a tananyag megértését.

Az általános iskolákban nagy az érdeklődés a számítástechnikai program iránt. A számítógépeket örömmel fogadták a tanárok és a diákok is. Sok iskola kért már a program indulása előtt számítógépet — ki ingyen, ki pénzt is hajlandó áldozni rá. Ezeknek az igényeknek 1985-ben csak kb. egyharmad részét lehetett kielégíteni. Annak feltétele, hogy a következő években folytathassuk az iskolák számítógéppel való ellátását az, hogy ezek a gépek tömegesen beszerezhetők legyenek. Ezért a TII pályázatot írt ki az általános és középiskolai oktatásra alkalmas személyi számítógépek szállítására. A pályázat eredménye 1986 márciusában várható.

Reméljük, hogy a központi erőfeszítések, a helyi lelkesedés és kezdeményezés együttesen az általános iskolákban is hamarosan olyan eredményre vezet, amilyet a középiskolák mutattak fel igen rövid idő alatt.

DR. SZŰCS BARNA
Országos Pedagógiai Intézet

ZSÁRY PIROSKA
Tudományszervezési és Informatikai Intézet

HT—1080Z

Hasznos gépi rutin

BASIC programok illetéktelen listázása, átírása ellen tökéletes védelmet nyújt az alábbi eljárás, amely futtatja a behívott programot, és érvényteleníti a BREAK, RESET gombokat már a betöltés alatt is. Egy rövid gépi kódú rutin kell hozzá, amely új, hasznos BASIC utasítások használatát teszi lehetővé:

SAVE "név"; kezdőcím, bájtok száma
— tetszőleges tárterület kimentése kazettára, SYSTEM-mel vagy LOAD-dal olvasható
RSET — reset, break érvénytelenítés
LSET — SYSTEM parancs letiltása
LOAD— tetszőleges SYSTEM formátumú vagy SAVE utasítással kiírt fájl betöltése. Név nem adható meg.

A rutin beírása

Írjuk be a következő programot:

```
Ø REM
1 CLEAR
2 POKE 16792,171:POKE 16739,67:
  POKE 16795,165:POKE 16796,67
3 POKE 168Ø1,238:POKE 168Ø2,66:
  POKE 16777,1Ø1:POKE 16778,67
```

Ügyeljünk arra, hogy a Ø-ás sor 205 darab szóközt tartalmazzon! Ezután a mellékelt hexadecimális listát gépeljük be monitor módban a 42EE hexadecimális címtől kezdve. A programot futása után vegyük fel kazettára. Úgy használhatjuk, hogy mögé gépeljük a szükséges BASIC programot, vagy kazettáról mögé töltjük. A programot minden használat előtt futtassuk!

Hexadecimális programlista

```
CD 84 02 3E 55 CD 64 02 CF 22 06 06 7E FE 22 23
28 09 CD 64 02 10 F5 CF 22 18 07 3E 20 CD 64 02
10 FB CF 3B CD 02 2B D5 CF 2C CD 02 2B C1 E5 60
69 18 24 06 FF 3E 3C CD 64 02 78 CD 64 02 7D CD
64 02 4F 7C CD 64 02 81 4F 7E CD 64 02 81 4F 23
1B 10 F6 79 CD 64 02 7A B7 20 D8 B3 28 03 43 18
D4 3E 78 CD 64 02 AF CD 64 02 3E 66 CD 64 02 E1
C9 21 E8 42 C3 1E 1D E5 CD 93 02 CD 35 02 FE 55
20 F9 CD 2C 02 CD 35 02 FE 78 28 23 FE 3C 20 F5
CD 35 02 47 CD 14 03 85 4F CD 35 02 77 23 81 4F
10 F7 CD 35 02 B9 28 DA 3E 45 32 3E 3C 18 D6 CD
14 03 CD F8 01 E1 C9 E5 21 AC 41 18 04 E5 21 E2
41 36 C3 23 36 5F 23 36 43 E1 C9
```

Öninduló BASIC program kimentése

LSET:SAVE"név"; 16384,
PEEK(16633)+PEEK(16634)*256−16383

Figyelem! Az LSET utasítás végrehajtása után nem használhatjuk a SYSTEM utasítást, amíg be nem írjuk: POKE 16866,2Ø1.

Ha azt akarjuk, hogy az öninduló program megállíthatatlan is legyen, így mentsük ki:

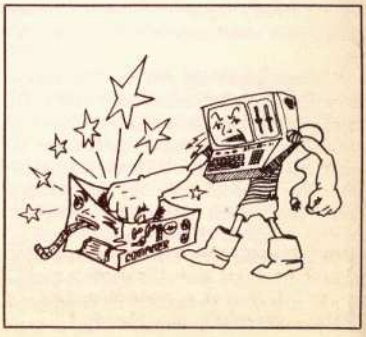
RSET:LSET:SAVE"név"; 16384,
PEEK(16633)+PEEK(16634)*256
−16383

Az így „kezelt" programok semmilyen módon nem állíthatók meg, még END, STOP utasítás, program vége, hibajelzés esetén sem állnak le. Ilyenkor a programfutás az elejétől indul. Ha mégis be akarunk építeni megállítási lehetőséget, például megfelelő kulcsszó beírása esetén, a következő utasításssort helyezzük el a programban: POKE 16812,2Ø1:END

A képernyőtartalom rögzítése

SAVE"név"; 15360,1Ø24
LOAD-dal programból olvasható.

UHERKOVICH PÉTER



A MINŐSÉGÜGY KÖZÜGY

A nyersanyagoknak és a termékeknek a minősége napjainkban újra a legfontosabb kérdések között van.

A számítástechnikában a minőség problémája már a kezdet kezdetén az érdeklődés előterébe került, és azóta sem tudta onnan semmi más kiszorítani. Ezt a különleges állapotot a számítástechnikában egyedül az állandó és erős verseny hozta létre, tartotta és tartja fenn.

A hardver, a szoftver és a szolgáltatások minőségével és a minősítés gyakorlatával minden aktív számítástechnikus naponta elkerülhetetlenül kapcsolatba kerül.

A minőség tehát a számítástechnikus munkájában megszokott és közönséges tényező. Nem új, nem rendkívüli és nem elkülönült a vele való foglalkozás, hanem természetes, és mindenben szerepet játszó.

Ez a kétségtelen tény azt a látszatot keltheti, hogy a számítástechnikus valamiféle különleges minőségi lény, és a számítástechnika területén a minőséget szemmel tartják és kézben tartják, és így vele kapcsolatban minden rendben van. Ez azonban nem egészen igaz. Mind az előttünk járó külföldön, mind pedig itthon, más-más okokra visszavezethetően, a kétségtelen pozitívumok mellett a kelleténél több a nem szükségképpeni negatívum.

Különösen az újabb időkben a hazai minőségi színvonal az elvárhatóhoz és az elérhetőhöz képest nem fejlődött eléggé, sőt több helyen az eltűrhető mértéket meghaladóan vissza is esett. Vonatkozik ez mind a hardver, mind a szoftver, mind a szolgáltatások, sőt az oktatás területére is. Kirívó minőségromlással is járó morális színvonalcsökkenés is tapasztalható.

Nem elégedhetünk meg tehát azzal, amit eddig csináltunk a minőségügy terén. Szélesebb fronton, lényegesen nagyobb erőkkel és hatékonyabban kell nekünk is munkához látni, ugyanúgy, ahogyan az országos minőségügyi problémákat a többi területen kézbevesszük.

Hiba volna a minőségügyet olyan időszakos nekibuzdulásként felfogni, amit most ránk kényszerít az élet, de majd csak megszabadulunk tőle valahogyan. A minőséget gondolkodásmódunk meghatározó szerepű tényezőjévé kell tennünk. Tudatosítanunk kell, hogy nemcsak ipari és mezőgazdasági nyersanyagnak, termékeknek és szolgáltatásnak van minősége, hanem minden másnak, pl. hangversenynek, tévéműsornak, kulturális és egészségügyi szolgáltatásnak is. És ezek összefüggnek és hatnak egymásra.

Miktől függ a minőség? Anyagi adottságoktól, a mi adottságainktól kétségtelenül. Ezeket azonban nem tudjuk lényegesen befolyásolni. Függ azonban mástól is. A mi viselkedésünktől, attól is például, hogy hogyan bánunk az adottságokkal, a nyersanyagokkal, a termelőeszközökkel és egymással is. És ez az, ami a fontosabb. Ez határozza meg nagyobb mértékben az egyén és a társadalom életének a minőségét. Ha pedig így, *az élet minősége* oldaláról nézzük a minőségügyet, máris látszik, hogy ez több mint egy ügy a sok közül.

Az élet minősége iránti felelősségérzet, de a közvetlen érdekünk is megköveteli, hogy sajátos eszközeinkkel támogassuk az országos erőfeszítéseket, és emellett a saját területünkön is indítsuk meg s körültekintő előrelátó és szigorú munkát a minőségügy rekonstrukciójára.

Ezt a tevékenységet egy számítástechnikai magazin — lehetőségei által meghatározottan — két területen művelheti sikerrel. Az egyik a számítástechnika hasznosítása a minőségügyi munkában, a másik pedig a számítástechnika saját belső, minőséggel kapcsolatos tennivalóiban való részvétel. Igyekezni fogunk, hogy mindkettőből kivegyük részünket. Az utóbbi feladatkörben — adottságainkból kifolyólag — a publikáció, a szakmai közlés minősége iránt is különleges felelősséget kell hogy érezzünk. És e téren is van javítanivaló bőségesen.

A jó példák mellett a hibákból is sokat lehet tanulni. Az utóbbi időben, sajnos, ez a második tanulási lehetőség már már bővebb volt, mint az előbbi. Ennek visszaszorítása érdekében nem lehetünk elnézők sem a mi magunk, sem a mások által közzétett tévedésekkel, hibákkal, hiányokkal szemben.

„Zöldségeskert" rovatunkban a szakmai zöldségeket, nyelvi és szakmai gyomokat tesszük közzé — ha kell, megjegyzések kíséretében. Természetesen csak azokkal foglalkozhatunk, amelyeket észreveszünk, vagy amelyek tudomásunkra jutnak, és tanulságosak is.

Zöldségeskert

„A japán Sharp cég PC 5000 elnevezésű, zsebre vágható kalkulátora is díjat kapott a személyi számítógépek nemes versengésében" . . . „A zsebbe csúsztatható, de bonyolultabb műveletek elvégzésére is alkalmas gépek közül a Sharp PC 5000-es vitte el a pálmát." (Kristály Magazin, a Pajtás különkiadványa. 1985/1.)

És a gép mindössze 4,3 kg.

„Karl Weierstrass (1815—1897) tétele szerint minden folytonos görbe tetszőleges pontossággal közelíthető polinommal." (Számítástechnika. Gimnáziumi tankönyv. Bp., 1985. Tankönyvkiadó.)

Az n·sin(x) függvény folytonos, bármely polinomtól való eltérése legalább n. Tehát e függvény csak tetszőleges pontatlansággal közelíthető polinommal. És akad még néhány hasonló példa.

Weierstrass neve a pontossággal forrt össze. A „weierstrassi szigor" fogalom a matematikában. Úgy látszik, hogy a matematika professzorai már nemcsak a weierstrassi szigort felejtették el, hanem Weierstrass fundamentális tételét sem ismerik. Szegény Weierstrass! Mi mindent fognak majd még neki tulajdonítani?

„SINCLAIR QL. Ez a kis gép már sok olyan tulajdonsággal rendelkezik, amelyek a professzionális gépekre jellemzők. 32 K-s tárkapacitása lehetővé teszi, hogy a szorzást és az osztást egyetlen műveletben elvégezze." (Technika Magazin 1986. Megjelent 1985 októberében.)

Sir Clive Sinclair azonban még nem tud róla, hogy mire képes gépének 32 K-s tárkapacitása. Ne is mondjuk meg neki!

HT—BASIC-DATA ÚJBÓL

A μM 1985/2. számában Besenyei Péter írása a HT iskolaszámítógép BASIC-jének hiányosságára mutat rá, de nem hibájára, hiszen a gép a leírásoknak megfelelően működik. A cikk a DATA — REM utasítások átírását ajánlja célravezetőbb módszerként. E módszernek azonban több olyan hibája van, amelyek miatt nem javasolom a programba történő beépítését.

1. Ez a megoldás csökkenti a program áttekinthetőségét, mivel a program futása közben módosítja önmagát. Márpedig az áttekinthetőség egy magas szintű programnyelven írt programnak nagyon fontos követelménye.

2. A megjegyzés utasítás kétféle módon is tárolható. REM begépelése esetén úgy, ahogy a cikkben olvasható. Sokan szeretik azonban a REM-et rövidíteni (?) aposztróf begépelésével. Ez viszont 3 bájtot igényel a tárból, így a program tulajdonképpen hosszabb lesz. A leírt szubrutin akkor is problémásan működik, ha a REM elé véletlenül szóköz kerül.

3. A cikk írója azért készítette el szubrutinjait, mert az eredeti megoldás lassú. Jó lenne azonban ennél is gyorsabb megoldást találni!

4. Egy alkalmazáshoz két szubrutint is kell használni; feleslegesnek tűnő oda-vissza átírogatások történnek. Kérdés: mi van akkor, ha éppen a visszaírás előtt nyomjuk meg a BREAK vagy a RESET gombot, majd újraindítjuk a programot RUN-nal? A válasz egyértelmű: csőd!

5. A szerző feltételezte, hogy a program a 42E9 hexadecimális (17129 decimális) tárcímen kezdődik. Ez pedig nem mindig van így. Például az egyre gyakrabban használt BASIC III (LEVEL 3) esetében ez a kezdőcím megváltozik, és más eset is elképzelhető. Ezért ezt a címet nem ajánlatos fixnek tekinteni.

Ezek után egy olyan megoldást közlök, amely az előző problémákat „kivédi". Tudjuk, hogy az iskolaszámítógép BASIC-je a READ utasításhoz szükséges pointert a 16639-16640-es memóriacímen tárolja.

Az *1. program* egy ciklusban kiolvassa minden adathoz ezt a pointert, majd az A tömbbe tárolja. Az *eredménylista* 1-2. oszlopa a 2 bájtos pointert tartalmazza, a 3. folyamatos sorszám, a 4. oszlopban pedig az utoljára beolvasott érték látható (J Ø-ról indul).

A program 14Ø-15Ø-es sorában beállítódik a READ pointere az A tömbben tárolt érték alapján. Érdekessége, hogy DATA utasítá-

```
5 REM   READ - RESTORE
6 REM
7 REM A HT gep a 16639-16640-es memoria
  cimen tarolja a READ utasitas pointeret.
8 REM
10 DATA 1
20 DATA 2
30 DATA 3
40 DATA 4
50 DATA 5
60 DATA 6,7
70 DATA 8,9
80 DATA 10,11
90 DIM A(14,1)
100 FOR I=1 TO 14
110 PRINT PEEK(16639),PEEK(16640),I,J
115 A(I,0)=PEEK(16639): A(I,1)=PEEK(16640)
120 READ J
130 NEXT I
140 POKE 16639,A(11,0)
150 POKE 16640,A(11,1)
160 READ J
170 PRINT ,,,J
180 B$=INKEY$: IF B$ = "" THEN GOTO 180
2000 DATA 14,15
2010 DATA 16
```

1. program

Az 1. program eredménylistája

232	66	1	0
100	67	2	1
108	67	3	2
116	67	4	3
124	67	5	4
132	67	6	5
140	67	7	6
142	67	8	7
150	67	9	8
152	67	10	9
161	67	11	10
164	67	12	11
124	68	13	14
127	68	14	15
			11

```
5 REM  READ - RESTORE n
6 REM
7 REM A C$ valtozoba egy gepi kodu rutin
  kerul, amely kikeresi a megfelelo sort.
8 REM
10 FOR I=1 TO 20
20 READ J
30 C$=C$+CHR$(J)
40 NEXT I
50 DATA 213,197,245,205,127,10,229,
   209,205,44,27,197,225,43,241,193,
   209,195,154,10
60 C=VARPTR(C$)
65 REM A gepi kodu rutin kezdocimenek
   beallitasa.
70 POKE 16526,PEEK(C+1)
80 POKE 16527,PEEK(C+2)
90 CLS
110 DATA 100,200
120 DATA 300,400
130 DATA 500,600
200 READ K,L
210 PRINT K,L
215 REM RESTORE 50
217 A=USR(50)
220 POKE 16639,A-INT(A/256)*256
230 POKE 16640,INT(A/256)
240 READ K,L
250 PRINT K,L
255 REM RESTORE 130
260 A=USR(130)
270 POKE 16639,A-INT(A/256)*256
280 POKE 16640,INT(A/256)
290 READ K,L
300 PRINT K,L

 100              200
 213              197
 500              600
READY
```

2. program

son belüli bármely adatra is beállítható, ezután a READ-del folyamatosan lehet olvasni. Ennek a megoldásnak az a hátránya, hogy végig kell olvasni előbb az összes adatot, de nem függ a DATA sor sorszámától!

Egy másik megoldást mutat a *2. program*, amelynek az a feladata, hogy kikeresse a megadott sorszámú sort. A gépi kódú program a C8 nevű változóba fog kerülni (10—40-es sor; a gépi kódú rutint az 50-es sor tartalmazza). Ezzel a megoldással a memória felső határát nem kell megváltoztatni. A C változó a C8 címét tárolja (60-as sor). A 16526-16527-es címre beáll a gépi kódú rutin kezdőcíme (70—80-as sor). Ebből látható, hogy valójában C egy pointer címét tartalmazza. A 217-es sor hatására kikeresi az 50-es sor címét (50-es sor címe—1), majd az előzőek alapján (220—230-as sor) beállítódik a READ pointere. Később hasonlóan lesz beállítva a 130-as sor DATA-ja. A program után a képernyőn megjelenő néhány sor látható.

A gépi kódú rutin tartalmazza a CALL 1B2CH utasítást. A ROM-ban található kis programrész megadja a kívánt BASIC sor kezdőcímét. A flagek: NC/Z — nincs ilyen sorszám; a megadott szám túl nagy. NC/NZ — nincs ilyen sorszám; a következő létezőre áll be. A belső rutin az összes regisztert használja, ezért vannak a regisztermentések. A rutin nem végez hibavizsgálatot!

A *3. programban* a gépi kódú rutin végzi el a READ utasítás pointerének beállítását is. Míg az előző esetben a rutin esetleg más hasonló célra is felhasználható, ebben a formájában kifejezetten a RESTORE n funkciót látja el.

Nem feltétlenül szükséges a regiszterek mentése, így a rövidített változat is megfelel (*4. program*). Az A=USR(50) hatására máris megtörténik a kívánt sor kikeresése és a pointer beállítása. Az

3. program

```
 5 REM  READ - RESTORE n     2.
 6 REM
 7 REM A C$ valtozoba egy gepi kodu rutin
   kerul, amely kikeresi a megfelelo sort.
 8 REM
 10 FOR I=1 TO 23
 20 READ J
 30 C$=C$+CHR$(J)
 40 NEXT I
 50 DATA 213,197,245,205,127,10,229,
    209,205,44,27,197,225,43,34,255,64,
    241,193,20, 9,195,154,10
 60 C=VARPTR(C$)
 65 REM A gepi kodu rutin kezdocimenek
    beallitasa.
 70 POKE 16526,PEEK(C+1)
 80 POKE 16527,PEEK(C+2)
 90 CLS
 110 DATA 100,200
 120 DATA 300,400
 130 DATA 500,600
 200 READ K,L
 210 PRINT K,L
 215 REM RESTORE 50
 217 A=USR(50)
 240 READ K,L
 250 PRINT K,L
 255 REM RESTORE 130
 260 A=USR(130)
 290 READ K,L
 300 PRINT K,L

  100              200
  213              197
  500              600
READY
>_
```

```
 REM  READ - RESTORE n     3.
 REM
 REM A C$ valtozoba egy gepi kodu rutin
 kerul, amely kikeresi a megfelelo sort.
 REM
 0 FOR I=1 TO 17
 0 READ J
 0 C$=C$+CHR$(J)
 0 NEXT I
 0 DATA 205,127,10,229,209,205,44,27,197,
   225,43,34,255,64,195,154,10
 0 C=VARPTR(C$)
 5 REM A gepi kodu rutin kezdocimenek
   beallitasa.
 0 POKE 16526,PEEK(C+1)
 0 POKE 16527,PEEK(C+2)
 0 CLS
 10 DATA 100,200
 20 DATA 300,400
 30 DATA 500,600
 00 READ K,L
 10 PRINT K,L
 15 REM RESTORE 50
 17 A=USR(50)
 40 READ K,L
 50 PRINT K,L
 55 REM RESTORE 130
 60 A=USR(130)
 90 READ K,L
 00 PRINT K,L
```

4. program

A változó melléktermékként az adott sor kezdő címe —1-et fog tartalmazni.

Amennyiben a program több gépi kódú rutint is tartalmaz, ügyelni kell, hogy a 70-80-as sorban található POKE utasítások segítségével mindig a megfelelő rutin címe legyen beállítva.

A program áttekinthetősége érdekében a gépi kódú rutinhoz célszerű néhány REM utasítást elhelyezni.

Végül egy sajtóhibára hívom fel a figyelmet. A hivatkozott cikkben szereplő rutin 1030-as sorában ez áll:

. . .IF D=147 OR 136 THEN. . ., ami helyesen a következő:

IF D=147 OR D=136 THEN. . .

Az első esetben sem kapunk hibajelzést, de a gép másképpen működik.

MIKES GÁBOR

Futás közben változó programok

A POKE és PEEK utasításokat gyakran használjuk, de többnyire ügyelve arra, hogy a BASIC-program tárolására szolgáló területet ne változtassuk meg, hiszen ez a program megváltozásához vezetne. Néhány esetben viszont a program futás közbeni átalakításával olyan problémák is megoldhatók, amelyek más módon csak nehezen kezelhetők.

Ismert például, hogy a BASIC-fordítóprogramok (interpreterek) egy része nem tudja az INPUT utasítás végrehajtásakor a válaszként begépelt képletet értelmezni. Kivétel például a Sinclair ZX-Spectrum gépe. A BASIC-program tárolása ugyanis közbülső kódok (tokenek)segítségével történik, azaz a BASIC-kulcsszavak helyett a gép csak egy egybájtos kódot tárol. Az INPUT végrehajtásakor viszont az interpreter a közbülső kódra történő átalakítást nem végzi el, ezért ezt a programnak kell megoldania. Egy ilyen program működésének az alapja a következő:

1. A felhasználó által beírt képletben előforduló kulcsszavakat (SIN, EXP, ABS, SQR, +, *, /, = stb.) közbülső kódba kell alakítani.

2. Az így átalakított képletet egy erre a célra fenntartott, eredetileg valamilyen karakterrel teleírt sorba írjuk be bájtonként. A képlet utolsó bájtja után egy kettőspontot, ezután a RETURN utasítás tokenjét, végül egy másik kettőspontot kell betölteni.

3. A képletben szereplő független változóknak értéket kell adni.

4. Egy GOSUB utasítással, mely a fenntartott sorra adja át a vezérlést, kiszámítjuk a függő változó értékét.

5. Az így kiszámított érték a program jellegétől függően kiíratható vagy további számításhoz felhasználható.

Egy ilyen elven működő, HT—1080Z gépre írt program látható az *1. listán.* A gép először bekéri az adott függvény képletét, majd egyenként rákérdez a képletben előforduló független változók értékére, és a kiszámított függvényértéket kiírja a képernyőre. Ezután választani lehet, hogy új képlettel vagy az előzővel számolunk ki újabb értékeket.

A program lehetővé teszi, hogy a független változók értékét (például trigonometrikus függvényeknél) fokban vagy PI többszöröseként adjuk meg. Ilyenkor az adott érték után a FOK, illetve a *PI karaktersorozatot kell beírni.

A program beírásánál ügyeljünk arra, hogy a 10—80-as sorokat karakterhelyesen gépeljük be, ellenkező esetben a POKE utasítások nem a megfelelő helyre töltenek. A program a beírt képletet a 40 sorba tölti, itt eredetileg csupa kettőspont van. Az első kettőspont címe 17 400. Az eredmény kiírása a 80-as sorban történik. A függő változó eredetileg Y, de ezt a program szükség esetén felülírja. A sor végén található karakter (Y) címe 17 700.

A *2. listán* egy ugyanilyen elven működő rajzolóprogram van. Ez is HT—1080Z gépre készült, de a változók nevei itt adottak. A független változót csak X, a függő változót csak Y jelölheti.

A program a kezelő által beírt függvény grafikonját rajzolja fel a megadott intervallumban. Az intervallumhatárok megadásánál itt is lehetőség van a FOK vagy a *PI használatára. A hibakezelő rutin lehetővé teszi, hogy olyan függvényeket is ábrázoljunk, amelyek bizonyos helyeken nincsenek értelmezve (például Y=1/X vagy Y=SIN(X)/X függvények). A közbülső kódba átalakított függvény itt is a 40-es sorba kerül, és ezt a sort hívja meg a rajzolást végző ciklus (770—830-as sorok).

ZÁTONYI SÁNDOR

1. lista

```
10 GOTO100
20 +-----------------------------------------+
   I          B E - E . V E T T E S I T O     I
   +-----------------------------------------+
30 KESZITETTE: Z A T O N Y I   S A N D O R BE.ESCSABA.1985.
40 ::::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::
:
50 CLS
60 PRINT"A FUGGVENY ERTEKE:"
70 PRINT
80 PRINTCHR$(PEEK(17400));" = ";Y
90 RETURN
100 DIMDEL(26):DIMVA(26):DIMVA$(26)
110 CLS
120 PRINT"KEREM A FUGGVENYT MEGADO KEPLETET, y=f(x)
   ALAKBAN."
130 PRINT
140 PRINT"A KEPLETET A BASIC SZABALYAI SZERINT KELL BEIRNI.
150 PRINT"A FUGGVENYBEN TOBB VALTOZO IS SZEREPELHET."
160 PRINT"A VALTOZOKAT EGY-EGY NAGYBETUVEL KELL JELOLNI."
170 PRINT@960,"A KEPLET BEIRASA UTAN NYOMD LE A (NEW LINE)
   GOMBOT!";
180 PRINT@512,"";
190 INPUTKEP$
200 PRINT@512,"";
210 FORII=1TOLEN(KEP$)
220     PRINT" ";
230     FG$=MID$(KEP$,II,3)
240     IFFG$="SGN"THENFG=215:GOTO350
250     IFFG$="INT"THENFG=214:GOTO350
260     IFFG$="ABS"THENFG=217:GOTO350
270     IFFG$="SQR"THENFG=221:GOTO350
280     IFFG$="LOG"THENFG=223:GOTO350
290     IFFG$="EXP"THENFG=224:GOTO350
300     IFFG$="COS"THENFG=225:GOTO350
310     IFFG$="SIN"THENFG=226:GOTO350
320     IFFG$="TAN"THENFG=227:GOTO350
330     IFFG$="ATN"THENFG=228:GOTO350
340     GOTO400
350         POKEII+17399,FG
360         POKEII+17400,32
370         POKEII+17401,32
380         II=II+2
390         GOTO520
400     KAR=ASC(MID$(KEP$,II,1))
410     IFKAR=43THENKAR=205:GOTO510
420     IFKAR=45THENKAR=206:GOTO510
430     IFKAR=42THENKAR=207:GOTO510
440     IFKAR=47THENKAR=208:GOTO510
450     IFKAR=91THENKAR=209:GOTO510
460     IFKAR=61THENKAR=213:GOTO510
470     IFKAR=40ORKAR=41ORKAR=46THEN510
480     IFA7(KARANDKAR(58THEN510
490     IFKAR(65OR90(KARTHENRUN
500     DEL(KAR-64)=1
510         POKE17399+II,KAR
520 NEXT
530 POKE17399+II,58
540 POKE17399+II+1,146
550 POKE17399+II+2,58
560 CLS
570 PRINTKEP$
580 PRINT"KEREM A VALTOZOK ERTEKEIT. "
590 PRINT@128,"";
600 FORII=1TO26
610     IFDEL(II)(1THEN690
620     IFPEEK(17400)=II+64THEN690
630     PRINTCHR$(II+64);" = ";
640     INPUTVA$(II)
650     VA=VAL(VA$(II))
660     IFRIGHT$(VA$(II),3)="FOK"THENGOSUB830
670     IFRIGHT$(VA$(II),3)="*PI"THENVA=VA*3.14159265
680     VA(II)=VA
690 NEXT
700 A=VA(1):B=VA(2):C=VA(3):D=VA(4):E=VA(5):F=VA(6):G=VA(7)
710 H=VA(8):I=VA(9):J=VA(10):K=VA(11):L=VA(12):M=VA(13)
720 N=VA(14):O=VA(15):P=VA(16):Q=VA(17):R=VA(18):S=VA(19)
730 T=VA(20):U=VA(21):V=VA(22):W=VA(23):X=VA(24):Y=VA(25)
740 Z=VA(26)
750 GOSUB840
760 POKE17700,PEEK(17400)
770 GOSUB850
780 PRINT@960,"AKARSZ MEG EZZEL A KEPLETTEL SZAMOLNI?
   (I/N) ";
790 V$=INKEY$:IFV$=""THEN790
800 IFV$="I"THEN560
810 IFV$="N"THENRUN
820 GOTO790
830     IFVA)36000THENVA=VA-36000:GOTO830
840     IFVA)3600THENVA=VA-3600:GOTO840
850     IFVA)360THENVA=VA-360:GOTO850
860     IFVA(-36000THENVA=VA+36000:GOTO860
870     IFVA(-3600THENVA=VA+3600:GOTO870
880     IFVA(0THENVA=VA+360:GOTO880
890     VA=VA*0.0174532925
900     RETURN
```

2. lista

```
10 GOTO100
20 +-----------------------------------------------------+
   I      F U G G V E N Y G O R B E  -  R A J Z O L O    I
   +-----------------------------------------------------+
30 KESZITETTE: Z A T O N Y I  S A N D O R BEKESCSABA, 1984.
40 ::::::::::::::::::::::::::::::::::::::::::::::::::::::
   ::::::::::::::::::::::::::::::::::::::::::::::::::::::
   ::::::::::::::::::::::::::::::::::::::::::::::::::::::
   ::::::::::::::::::::::::::::::::::::::::::::::::::::::
   :
50 IFERL=40ANDERR=2THENPRINT"HIBAS KEPLET :  ";KEP$:STOP
60 IFERL=40AND(ERR=6ORERR=100RERR=20)THENRESUME820
70 PRINT"HIBAS SOR :  ";ERL
80 PRINT"HIBAKOD :  ";ERR
90 STOP
100 ONERRORGOTO50
110 CLS
120 PRINT"KEREM A FUGGVENYT MEGADO KEPLETET, Y=F(X)
    ALAKBAN."
130 PRINT
140 PRINT"A KEPLETET A BASIC SZABALYAI SZERINT KELL BEIRNI."
150 PRINT@960,"A KEPLET BEIRASA UTAN NYOMD LE A (NEW LINE)
    GOMBOT!";
160 PRINT@512,"";
170 INPUTKEP$
180 PRINT@512,"";
190 FORII=1TOLEN(KEP$)
200     PRINT" ";
210     FG$=MID$(KEP$,II,3)
220     IFFG$="SGN"THENFG=215:GOTO330
230     IFFG$="INT"THENFG=216:GOTO330
240     IFFG$="ABS"THENFG=217:GOTO330
250     IFFG$="SQR"THENFG=221:GOTO330
260     IFFG$="LOG"THENFG=223:GOTO330
270     IFFG$="EXP"THENFG=224:GOTO330
280     IFFG$="COS"THENFG=225:GOTO330
290     IFFG$="SIN"THENFG=226:GOTO330
300     IFFG$="TAN"THENFG=227:GOTO330
310     IFFG$="ATN"THENFG=228:GOTO330
320     GOTO380
330        POKEII+17399,FG
340        POKEII+17400,32
350        POKEII+17401,32
360        II=II+2
370        GOTO490
380     KAR=ASC(MID$(KEP$,II,1))
390     IFKAR=43THENKAR=205:GOTO480
400     IFKAR=45THENKAR=206:GOTO480
410     IFKAR=42THENKAR=207:GOTO480
420     IFKAR=47THENKAR=208:GOTO480
430     IFKAR=91THENKAR=209:GOTO480
440     IFKAR=61THENKAR=213:GOTO480
450     IFKAR=40ORKAR=41ORKAR=46THEN480
460     IFA7<KARANDKAR<58THEN480
470     IFKAR<650R90<KARTHENRUN
480        POKE17399+II,KAR
490 NEXT
500 POKE17399+II,58
510 POKE17399+II+1,146
520 POKE17399+II+2,58
530 CLS
540 PRINT"KEREM AZ INTERVALLUMOK VEGPONTJAIT."
550 PRINT@384,"X TENGELYEN :"
560 PRINT@512,"";
570 INPUT"KEZDOPONT";KX$
580 INPUT"VEGPONT  ";VX$
590 PRINT@704,"Y TENGELYEN : "
600 PRINT@832,"";
610 INPUT"KEZDOPONT";KY$
620 INPUT"VEGPONT  ";VY$
630 KX=VAL(KX$):VX=VAL(VX$)
640 KY=VAL(KY$):VY=VAL(VY$)
650 IFRIGHT$(KX$,3)="FOK"THENKX=KX*0.0174532925
660 IFRIGHT$(VX$,3)="FOK"THENVX=VX*0.0174532925
670 IFRIGHT$(KY$,3)="FOK"THENKY=KY*0.0174532925
680 IFRIGHT$(VY$,3)="FOK"THENVY=VY*0.0174532925
690 IFRIGHT$(KX$,3)="*PI"THENKX=KX*3.14159265
700 IFRIGHT$(VX$,3)="*PI"THENVX=VX*3.14159265
710 IFRIGHT$(KY$,3)="*PI"THENKY=KY*3.14159265
720 IFRIGHT$(VY$,3)="*PI"THENVY=VY*3.14159265
730 IFVX<=KXORVY<=KYTHEN530
740 CLS
750 DX=(VX-KX)/127
760 II=0
770 X=KX+II*DX
780 GOSUB840
790 Y=(VY-Y)*47/(VY-KY)
800 IFY<00RY>47THEN820
810 SET(II,Y)
820 II=II+1
830 IFII<128THEN770
840 PRINT@0,KEP$;
850 PRINT@960,"AKARSZ MEG EZZEL A KEPLETTEL SZAMOLNI?
    (I/N)";
860 V$=INKEY$:IFV$=""THEN860
870 IFV$="I"THEN530ELSEIFPEEK(17347)<>90THENDELETE30-1000:'
880 IFV$="N"THENRUN
890 GOTO86
```

SZÁMALK SZAKKÖNYV

ISMERJE MEG,
használja fel sokoldalúan a számítástechnika korszerű eszközét,
A MIKROGÉPET!

A SZÁMALK könyvei a kezdő és gyakorlott felhasználóknak sok új, hasznos ismeretet kínálnak ehhez. Szíves figyelmébe ajánljuk a következő könyveinket:

A ZX—Spectrum programozása
200 oldal **84,— Ft**

A szerzők szakavatott vezetésével a számítástechnikával most ismerkedők a gép üzembe helyezésétől a programozásig juthatnak el.

Bodor Tibor—Gerő Péter:
A COMMODORE—64 programozásának gyakorlata
160 oldal **55,— Ft**

Gépkezelési és használati ismeretek alapozzák meg a programozási alapelvek ismertetését s a könyvben közölt, kipróbált programok futtatását.

Dr. Rózsa Lajos szerkesztésében:
Mikroszámítógép-alkalmazási esettanulmányok
372 oldal **156,— Ft**

Az ipari mérésadatgyűjtési és irányítási feladatok ellátására az MTA SZTAKI kifejlesztette a Mikroszámítógépes Folyamatirányító Berendezést (MFB), amelynek alapos ismertetése után hat konkrét feladat megoldása mutatja be sokoldalú felhasználhatóságát.

Marschik Iván:
Mikrogéprendszerek tervezése I—II.
768 oldal **439,— Ft**

Az első kötet — Rendszertechnika — a mikrogéprendszerek elemeit ismerteti (8 és 16 bites mikroprocesszorok, memóriák, beviteli/kiviteli elemek, sínrendszerek, mikroperifériák), a második kötet — Programrendszerek — pedig a felhasználói és rendszerfejlesztő programozással és operációs rendszerekkel (CP/M, MP/M, iRMX/80, MS—DOS, PC—DOS, UNIX), programnyelvekkel és fordítóprogramokkal foglalkozik részletesen.

A könyvek megvásárolhatók a SZÁMALK Könyvesboltjában: Budapest XI., Szakasits Árpád út 68., valamint a Művelt Nép és a Könyvterjesztő Vállalat boltjaiban.

SZÁMALK

Alapozás XIV.

Az állandó idővel késleltető operátor használatával sikerült egy-két jól működő matematikai modellt alkotnunk. Az információtárolásra alkalmas modelljeink működési tartománya azonban még nem volt kielégítő. Most a működési tartománynak, az alkalmazhatósági körnek a bővítésével foglalkozunk. Előbb azonban a közönséges késleltető működésére adunk egy jól használható szemléltetést.

Egy szemléletes közönséges késleltető modell

A késés és a késleltetés jelensége központi fontosságú mind az elméletben, mind a gyakorlatban. Sajnos — a tankönyvek és a szakkönyvek bizonyítják — az oktatás sehogyan sem, a „tudomány" pedig alig és akkor is mellékesen foglalkozik ezzel az alapjelenséggel. Fölösleges taglalni az olyan tudás értékét, gyakorlati alkalmazhatóságát, amely egy mindenütt lényeges szerepet játszó jelenséget nem vesz észre vagy nem tud vele mit kezdeni, és ezért másról beszél.

A késés és a késleltetés ilyetén kezelése azonban nem véletlen. Mindenki meggyőződhet róla, hogy nem könnyű területről van szó. A nehézség elsősorban lélektani okú. Nehéz ugyanis a folyamatok tudati követése, különösen, ha több van belőlük. (Hogy ennek a nehézségnek mik az okai, az más kérdés. Véleményünk szerint nem szükségképpeni nehézséggel állunk szemben, hanem félrevezetés és félrevezetődés eredményével. A lélektani okokkal e helyen nem foglalkozhatunk részletesebben, az Olvasó figyelmét azonban — úgy véljük — nem felesleges felhívni ezekre.)

A késés és késleltetés jelenségének megragadása azért nem szokott sikerülni, mert egy kaptafára kívánnak húzni minden késési jelenséget, holott a késés nem ragadható meg egyetlen egyszerű, egzakt modellel. Egy-egy késési jelenségre természetesen készíthetők egzakt leírások. Ezek a késési modellek azonban más késési jelenségek leírására már általában nem lesznek jók. Ez azonban nem ok arra, hogy az egész területet kezelhetetlenként semmibe vegyük.

Tudatosítsuk tehát, hogy az a késésmodell, amit a közönséges késleltetővel írtunk le, egyfajta késési jelenséget írt le, vannak azonban másfajta, a megismert módon meg nem ragadható, le nem írható késési jelenségek is. Később példát is fogunk látni ilyenekre.

A késési jelenségek helyes megkülönböztetnitudásában sokat segít, ha jó szemléletes modelljeink vannak. Egy ilyen szemléletes késleltetőmodellről lesz szó a következőkben, ami (módosítással) másfajta késési jelenségek modellezésére is alkalmassá tehető, most azonban csak a közönséges, állandó idejű késleltetés esetével foglalkozunk. Az *1. ábrán* egy regisztrálóberendezés vázlata látható. E regisztrálóberendezés valamilyen rendszer egyik jellemzőjének alakulását jegyzi fel, működéstörténetet készít. A hengerek egyenletes (egységnyi) kerületi sebességgel forognak.

A térben rögzített *B1* és *B2* pont összekötő szakaszán mozog az írótoll hegye. Közben a papír is halad, így az írótoll egy egyértékű függvény grafikonját rajzolja a papírszalagra.

A térben rögzített *K1* és *K2* pont összekötő szakaszán egy automatikus leolvasó figyeli és olvassa le a regisztrátumgörbének a leolvasó alá kerülő értékét.

Mi lesz a *K* leolvasó által leolvasott érték, ha a *B1B2* egyenes és a *K1K2* egyenes távolsága *a*? Nyilván *a* idővel előbb, a *B* írótoll által írott (feljegyzett) érték.

Tehát *B* helyzete és a regisztrátumgörbe *K* leolvasó alatti pillanatnyi „magassága" nem független egymástól.

Legyen *a* értéke *2* hosszúságegység!

Indítsuk el a regisztrálóberendezést a *0* időponttól! Nyilvánvaló, hogy legkorábban *a* idő elteltével tud *K* érdemi feljegyzést leolvasni.

Mi lesz a leolvasott *(k)* érték az *a* időpontban, azaz mi lesz *k(a)*? Nyilvánvalóan *k(2)=b(0).*

Azaz *K* által a *2* időpontban érzékelt érték megegyezik *B* által a *0* időpontban papírra tett (jegyzett) értékkel.

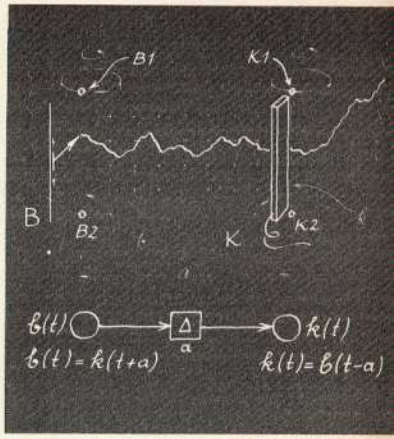
Ugyanígy igaz, hogy
k(3)=b(1),
k(4)=b(2),
k(5)=b(3).
Általában
$k(t)=b(t-2), \quad (t \geq 2),$
$k(t+2)=b(t), \quad (t \geq 0).$

A *b* tengelyen zajló mozgás pillanatnyi helyzetéről szóló információ késve, *a* késéssel jut el a *k* tengelyre.

A rendszer kapcsolási rajza az 1. ábra alján látható. A késleltetési időt az operátor jele alá írtuk.

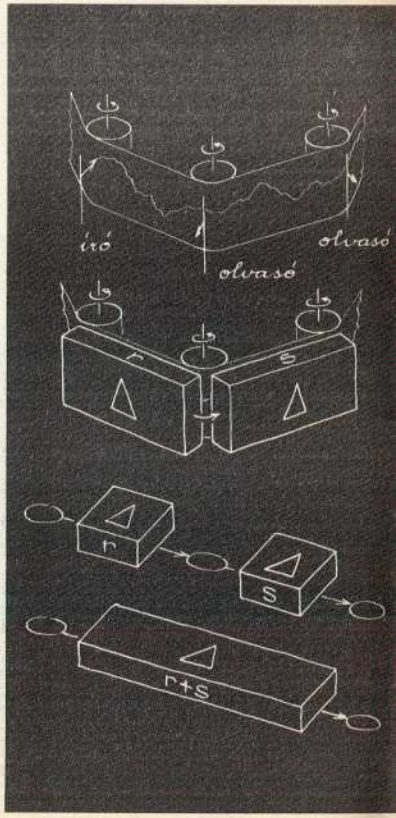
A papírszalagot továbbító hengerekre feszülő papír *b* és *k* közötti része szemlélteti a késleltető tartalmát. E modell a szó szoros értelmében betekintést enged a késleltető „belsejébe".

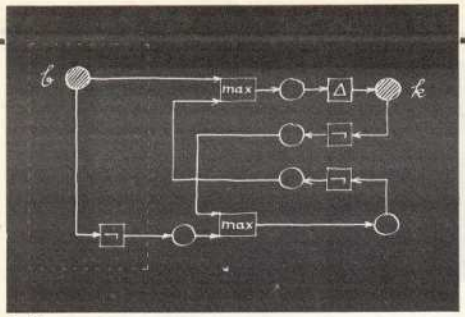
A *2. ábrán* az írófejet és az olvasófejet még az előzőkhöz képest is egyszerűbben ábrázoltuk. Az ábrákon jól látszik, hogy sorba kapcsolt késleltetők késleltetési idői összege lesz az eredő késleltetési ideje. Az eredő késleltető pillanatnyi tartalma pedig



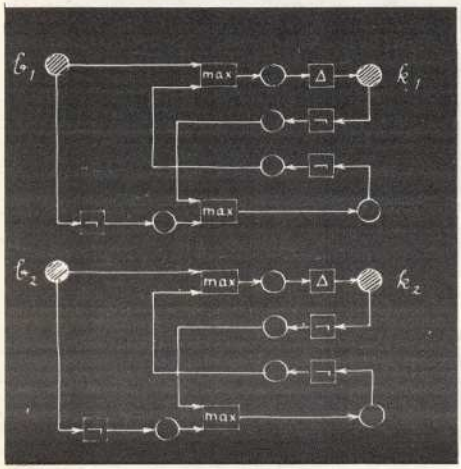
1. ábra

2. ábra

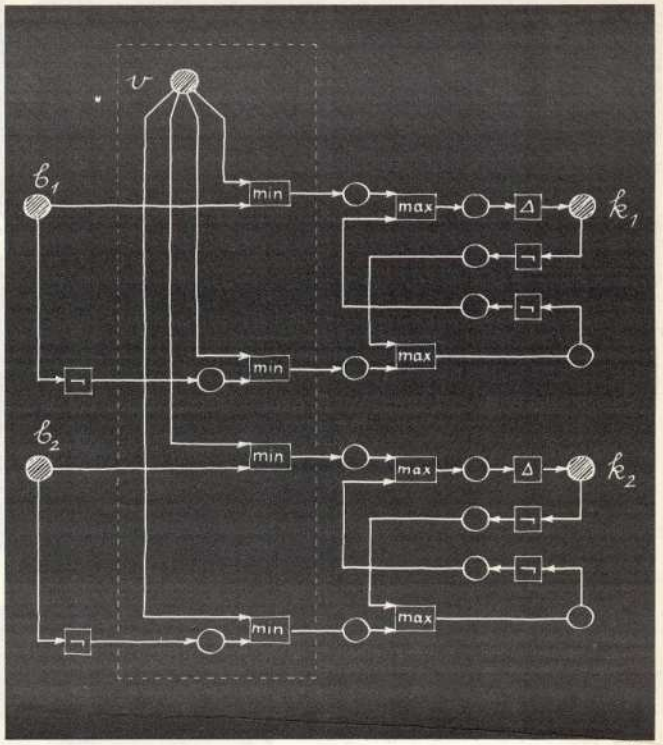




3. ábra



4. ábra



5. ábra

a két összetevő pillanatnyi tartalmának „egymás után helyezése" lesz.

A közönséges késleltető fogalmának szemléletes átismétlése után rátérünk fő feladatunkra, a tárolást elvégző operátorcsapatok kialakítására.

Az információtárolás folyamata

Információtároló operátorcsapatocskáink „megjegyeztek", „megtanultak" értékeket, azaz kimenetükön megjelent a megőrizni kívánt érték, és ez akkor is ott maradt, amikor a bemeneten már nem a megtanulandó információ volt. (Alapozás XIII.) Elemezzük ezt a jelenséget alaposabban!

Természetesen vetődik fel a kérdés, hogy honnan tudja a tárolócsapat, hogy a bemenetén levő információ mikor megtanulandó és mikor nem az, azaz olyan, amivel nem kell törődnie?

Az egyik operátorcsapat (Alapozás XIII. 2. ábra) olyan volt, hogy meg lehetett „tanítani" a $+1$ és a -1 értékre; ha elég hosszú idejű volt a „tanítás", a csapat „megjegyezte" és kimenetén fenntartotta a bemeneten tapasztalt értéket. Ezután a bemeneten széles körben (de nem teljesen szabadon) változhatott az érték, ezek az értékek nem befolyásolták a csapat munkáját, és a kimeneten fenntartott értéket sem.

A bemeneti folyamatokat (értékeket) tehát két osztályba sorolhatjuk aszerint, hogy azok a kimeneti értékeket képesek-e befolyásolni vagy sem. (Vannak olyan értékek, amelyek soha sem képesek a kimeneti állapot befolyásolására.) Ez a helyzet felfogható úgy is, hogy operátorcsapatunk magából a bemeneti folyamatból értesül arról, hogy van-e egyáltalán tárolni való, megtanulni való, és az éppen mi.

A másik operátorcsapat, amely a *0* és az *1* ill. az *1* és a *0* értékpár megjegyzésére volt használható (Alapozás XIII. 3. ábra) a kimenetén megtartotta ezeket a „megtanult" értékeket, akkor is pl., ha a bemenet értéke *0* és *0* értékpárra változott. A *0* és *0* értékpár nem befolyásolta sem az *1* és *0*, sem a *0* és *1* értékpárt mutató kimenetet. Ebből következik, hogy operátorcsapatunk a *0* és *0* értékpárt nem képes megjegyezni, hiszen ez az értékpár nem befolyásolja a kimeneti értékeket. Ha viszont a *0* és *0* megjegyeztetése is feladat, akkor két úton indulhatunk el. Vagy azon, amelyen eddig jártunk, amikor maga a bemeneti folyamat adta az információt arról, hogy most megjegyzendő érték(ek) van(ak) a kimeneten, vagy egy másik úton, amikor nem a bemeneti értékből, hanem máshonnan tudja meg az operátorcsapat, hogy a bemeneten éppen jelenlevő információ megjegyzendő-e vagy sem.

Mindkét út járható és fontos, de mivel az elsőre már láttunk példát, most a másodikkal foglalkozunk.

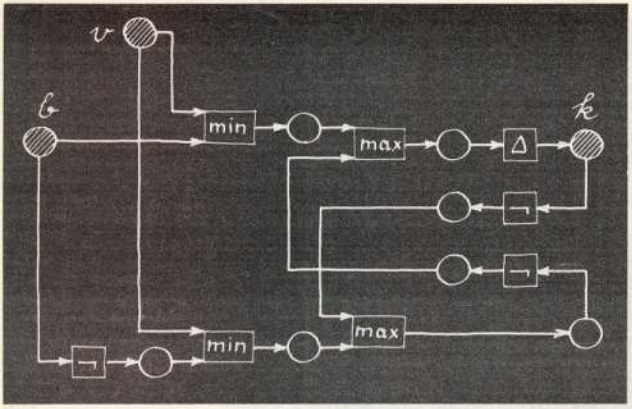
A tárolási parancs

Ha a tároló bemenetén szabadon, látszólag szabálytalanul váltogatják egymást az értékek, valamely bemeneti érték „megőrzésre kerülése" vagy maguktól az értékektől függ valamilyen módon, vagy attól, hogy a tároló operátorcsapat külön, más úton utasítást kap-e, hogy „Azt tárold, amit most észlelsz a bemeneteiden!".

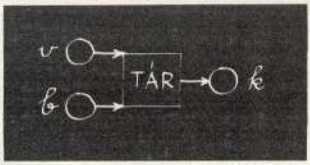
Lássunk egy példát erre az utóbbira!

A *3. ábrán* levő operátorcsapat alkalmas a *0* és *1*, ill. az *1* és *0* állapot tárolására (a rendszer kimenetein való rendelkezésre bocsátására, ill. azok rendelkezésre állásáról való gondoskodásra). Meg tudjuk oldani, hogy a bemeneteken csak *0* és *1*, ill. *1* és *0* jelenhessen meg (a pontozottan körülkerített rész révén).

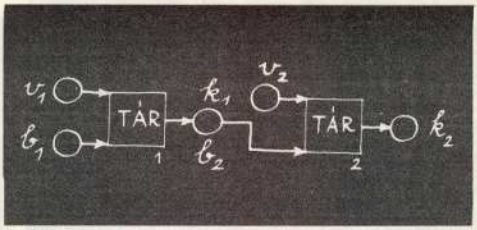
Ha két ilyen rendszert teszünk egymás mellé *(4. ábra)*, akkor az így kialakuló operátorcsapat már „megtanítható" nemcsak az eddigi *0* és *1*, az *1* és *0*, hanem a *0* és *0* meg az *1* és *1* értékpárra is.



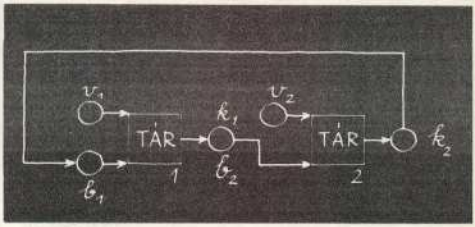
6. ábra



7. ábra



8. ábra



9. ábra

Próbáljuk ki! Legyen pl. $b1=0$ és $b2=0$ alkalmas ideig! Az eredmény $k1=0$ és $k2=0$ lesz. Rendszerünk tehát tanulékony.

Sajnos azonban ez az operátorcsapat nemcsak tanulékony, hanem feledékeny is. Ha a $b1=0$ és $b2=0$ állapot megszűnik, és utána pl. $b1=1$ és $b2=0$ állapot jelenik meg és áll fenn bizonyos ideig, akkor a $k1=0$ és $k2=0$ állapotból $k1=1$ és $k2=0$ állapot lesz.

A tárolási feladatot akkor tekintjük kielégítően megoldottnak, ha az információt nemcsak megtanulni, hanem — addig, ameddig kívánjuk, — megtartani is meg tudja az operátorcsapat.

A $k1=0$ és $k2=0$ állapot fennmaradását csak a $b1=0$ és $b2=0$ állapot fennmaradása biztosítja. Próbáljuk meg megóvni a bemeneteket attól, hogy bármelyikre is *1* kerülhessen, ha mi nem akarjuk!

A megóvási módszer (közismert nevén *kapuzás)* egy-egy minimumképző négy helyre való közbeiktatását jelenti. *(5. ábra)* Ha $v=0$, akkor „bárhogyan ugrál" is *b1* és *b2*, *k1* és *k2* értéke nem változik. A bemeneteken zajló folyamatok nem tudnak hatást gyakorolni a rendszerre, hiszen *„becsuktuk és csukva tartjuk előttük a kaput".* Ha viszont $v=1$, akkor „a kapu nyitva van", a bemenetek hatása szabadon és ugyanúgy érvényesül, mint ha az *1, 2, 3* és *4* jelű operátort nem is iktattuk volna a rendszerbe.

Hogyan működik tehát a tárolócsapat ezután a kiegészítés után?

❶ *b1*-re és *b2*-re a kívánt értékeket juttatjuk (ez a megtanulandó lecke).

❷ „Kinyitjuk a kaput", azaz gondoskodunk róla, hogy $v=1$ legyen.

❸ Gondoskodunk róla, hogy a „lecke" a kívánt ideig *b1* és *b2* bemeneteken legyen.

❹ „Becsukjuk a kaput", és gondoskodunk zárva tartásáról.

A kapu kinyitását tekinthetjük a tárolási folyamatot megindító parancsnak.

Nem nehéz belátni, hogy így nemcsak értékpárok, hanem értékhármasok sőt érték *n*-esek (*n* gyakorlatilag korlátlan) tárolása is megoldható.

Az értékegyes, azaz egyetlen „fúj" vagy „nem fúj" érték megőrzését végző tároló operátorcsapatocska a *6. ábrán* látható. Ezt a csapatocskát egyetlen operátornak is tekinthetjük, és külön jelölést alkalmazhatunk rá. *(7. ábra)*

Fontos tudatosítanunk, hogy sok másféle tárolófeladat ellátására alkalmas, hasznos operátorcsapat is van, ezek mindegyikére azonban itt nem lesz szükségünk.

A „TÁR" operátor olyan, hogy tárolási parancsra „tanul", és addig biztosan megjegyzi a megtanult értéket, ameddig újra tárolási parancsot nem adunk neki (azaz ki nem nyitjuk a kaput).

Ha a közönséges késleltetőt „feltöltjük valamilyen értékkel", akkor a késleltetési idő időtartamáig ez az érték lesz a kimenetén. Tehát a közönséges késleltető olyan tárolóelem, amely tanulásra képes, de csak adott időtartamra tudja megjegyezni a tanultakat. Sok esetben azonban nincs is többre szükség. Tudatosítsuk, hogy ez nem kevés, hiszen az adott ideig való tárolás tette lehetővé számunkra a korlátlan idejű tárolás megvalósítását is.

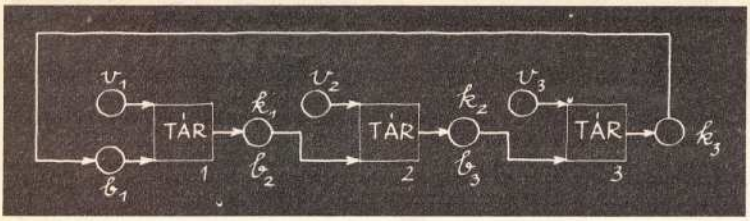
A tárelemek „lecke"-bemeneteire, azaz azokra a bemenetekre, amelyek a „tanulnivaló" folyamatokat fogadják, általában (de nem mindig) állandó folyamatokat szoktunk juttatni, legalábbis addig, ameddig „a kapu nyitva van".

A tárolás modellezésében jelentősen előrehaladva, a digitális berendezések működésének — talán legfontosabb — alapjelenségét is részletesebben tárgyalni tudjuk már. Mielőtt ezzel foglalkozni kezdenénk, fel kell hívnunk a figyelmet a tárolásproblléma különleges fontosságára. A következők megértéséhez nincs szükség az eddig megismert tárolófajtákon kívül más tárolók ismeretére. Ez azt a látszatot keltheti, hogy a tárolók nélkülözhetetlenek és fontosak ugyan, de területük nem túlságosan gazdag. Ez nem így van. Egy-egy digitális berendezés szellemessége, gyorsasága sokszor az elmés tárolásmegvalósításon múlik, ilyen ugyanis sokkal több van, mint amikkel itt foglalkozhattunk. És a sokféle konkrét építőelemmel is sokféleképp lehet berendezéseket létrehozni.

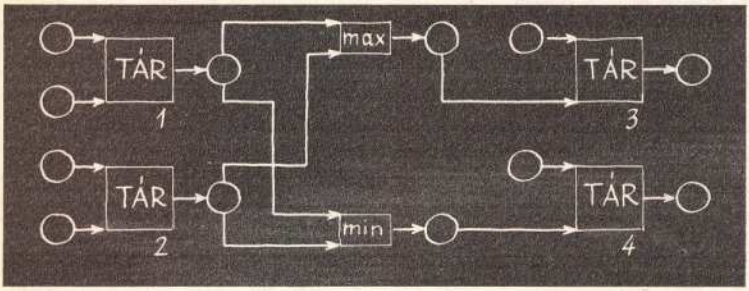
Nemcsak ez a három megtanulási-megjegyzési elv lehetséges, amit említettünk. Lehet például olyan tárolókat is szerkeszteni, amelyek hol tárolási parancsra tárolnak, hol pedig aszerint, hogy a tanulóbemenetükön milyen jelenségek zajlanak. Lehet aztán a tanulóbemeneten zajló folyamatokba pl. egyenlő időnként beékelni a tanulást tiltó vagy megparancsoló információkat; a bemenetet időosztással használja ilyenkor a tanulnivalót figyelő és a tanulási parancsot figyelő funkció.

A gép működése

Tudjuk, hogy a gépben tárolás, továbbítás és átalakítás folyik. E folyamatnak legkisebb megvalósulása szemlélhető a *8. ábrán.* Ha a kaput kinyitjuk, azaz gondoskodunk a $v2=1$ állapotról, akkor *k2* „átveszi"

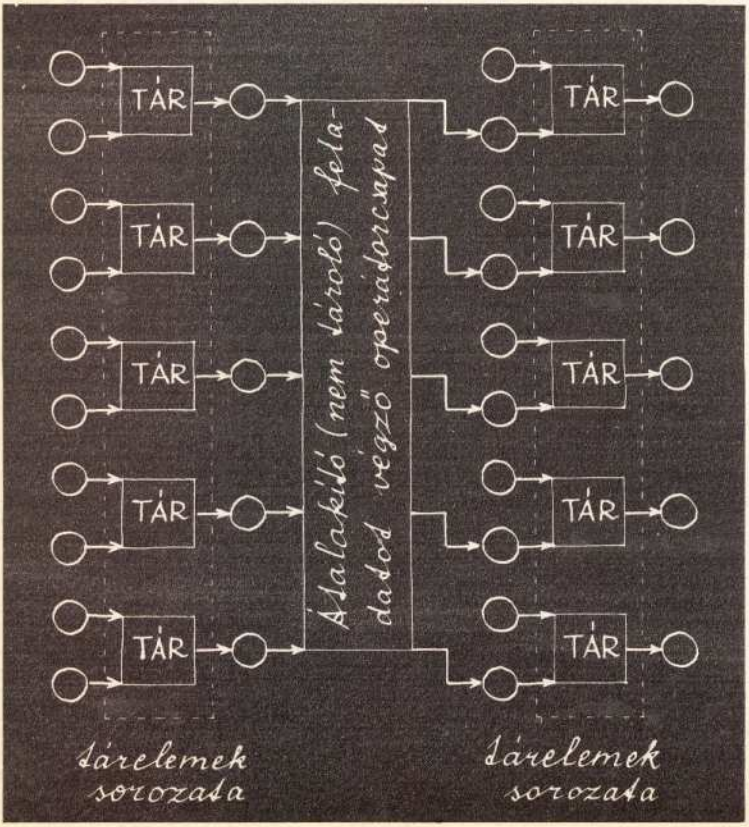


10. ábra



11. ábra

12. ábra



k1 értékét. Más szóval az *1*-es tár tartalma átmásolódik *(nem átmegy!)* a *2*-es tárba.

A *9. ábra* operátorcsapatában, ha egyszerre nyitjuk ki a kapukat, az *1*-es tár tartalma a *2*-esbe, a *2*-es táré az *1*-esbe íródik át. Ha valakinek ezzel kapcsolatban kételyei lennének, a részletes kapcsolási rajzból meggyőződhet, hogy ez tényleg így van. A kételkedő tehet azonban egy kivédhetetlen ellenvetést is. Nincs a gyakorlatban két pontosan egyenlő késleltetési idejű késleltető, így a 9. ábra rendszerének működése függ a késleltetési időktől és a kapuk nyitvatartási időitől. Noha folytatható lenne, mégsem folytatjuk e vitát. Két tár (az *1* és a *3* jelű) tartalmának felcserélésére biztosan jó a *10. ábra* operátorcsapata. Először a *v2* kaput nyitjuk ki. A *TÁR1* tartalmának a *TÁR2*-be másolódása után bezárjuk *v2*-t. Ezután kinyitjuk *v1*-et. Miután megtörtént a *TÁR3* tartalmának a *TÁR1*-be másolódása, ezt a kaput is bezárjuk. Végül pedig ugyanígy átmásoljuk a *TÁR2* tartalmát a *TÁR3*-ba.

E másolgatások természetesen nem változtatták meg az átmásolt információt. Ha a digitális berendezés csak erre volna képes, akkor csak információtároló gép volna. A tárolás és információforgalom lebonyolítása mellett azonban átalakításokat is tudunk a géppel végeztetni.

A *11. ábrán* látható kapcsolású operátorcsapat a *TÁR1* és a *TÁR2* tartalma maximumának ill. minimumának a *TÁR3*-ba, ill. a *TÁR4*-be írására képes. Ügyelni kell azonban arra, hogy a *v1, v2, v3* és *v4* kapuk ki- és becsukogatása hogyan és mikor történik. A kívánt eredmény elérése érdekében a maximumképző és a minimumképző műveletvégzési idejét, késését is figyelembe kell venni. A digitális berendezés legtipikusabb részletét a *12. ábra* mutatja. Az ilyen rész tárelemek tartalmát változatlanul vagy átalakítva tárelemekre írja át.

Lássunk néhány tipikus példát! Az átalakító operátorcsapattól „balra" két, 2-es alapú számábrázolási rendszerben felírt számot tartalmazó tárelemsorozat (regiszter) van, tőle „jobbra" pedig az eredmény fogadására és megőrzésére szolgáló tárelemsorozat (eredményregiszter). A szorzást végző részrendszer kapcsolási rajza ugyanilyen elvű, az átalakító rész természetesen eltér az előzőétől.

Egy szám szinuszát azonban — *noha lehetne* — más módon, több lépésben szokták kiszámítani. Ennek megismerésével azonban a gépépítés fejlettebb szakaszában foglalkozunk majd.

Mielőtt a gépépítés további egyszerű lépéseivel foglalkoznánk, a következőkben a vonaltulajdonságokkal dolgozó működésleírás és a késés finomabb elemzése kerül sorra.

POGÁNY CSABA

KIS GÉPEK — NAGY PROGRAMOK

A mikrogépek hatása a programozásra

A mikroelektronika gyorsan fejlődő világában 16 év már történelmi távlatot biztosít. Az első integrált áramkört 1959-ben jelentette meg a Texas Instruments, és ezzel elkezdődött a számítástechnikai eszközök nagymértékű miniatürizálásának folyamata. Az első integrált 4 bites mikroprocesszor, a 4004, az Inteltől származik, és 1971-ben látott napvilágot. Azóta a fejlődés jóformán töretlen, a mikroelektronika hatása életünk egyre több szférájában érezhető. A következőkben azt vizsgáljuk, hogyan befolyásolják a mikrogépek a programozás mesterségét, tudományát és művészetét.

Az előzmények

A harmadik generációs számítógépek elterjedésével együtt elindult egy elidenegenedési folyamat is. A programozók és a gépek közé egyre vastagabb és magasabb hardver- és szoftverfal épül. A be- és kiviteli csatorna, a processzorok, a megszakításkezelés, az operációs rendszerek a teljesítménynövelés ürügyén a felhasználók és a gép közé ékelődtek. A gépek teljesítménye valóban nőtt, a programozás azonban elvesztette, vagy legalábbis kezdte elveszteni egyik, most már tudjuk, hogy igen fontos tulajdonságát. Megszűnőben volt a programozás személyes volta, ami egyben a programozó egyéniségének megnyilatkozásait is lehetővé tette egy-egy program megírásakor. Akkor nem volt nyilvánvaló, de ma már az, hogy igazán jó programok csak az alkotó agymunkát végző ember és az ezt realizáló gépi eszköz közvetlen és zavartalan kapcsolatából születnek.

Amikor az IBM DOS és OS terjedelmes értelmetlen tolvajnyelven írott, ún. „job-control" programok előállítására kényszerítette a gépteremből már amúgy is kizárt programozókat — oda ugyanis csak külön köpenyben és gumipapucsban léphettek be az operátorok is —, egész természetes emberi reakciók születtek: a hobbiból és a hivatásból programozók jó része felhagyott a teljes programozással; a programozást iparként űzők megfelelő ideológia kidolgozásával vigasztalódtak; ahol a tehetség szerencsés környezeti adottságokkal párosult, ott új hardver- és szoftvereszközök létrehozásával igyekeztek visszaállítani a programozás „emberi arculatát".

A fenti események, és ezek néha előnyös, néha hátrányos következményei konkrét példákkal is illusztrálhatók. A 70-es évekre kialakult ún. szoftverkrízis jórészt azért állt elő, mert sok programozó kiábrándult a zárt rendszerben üzemelő gépóriások etetésévé fajult, az alkotás örömét egyre kevésbé nyújtó szakmájából.

A kötegelt feldolgozásban működő operációs rendszerek emellett azzal is bosszantották a programozókat, hogy a tényleges feladat szempontjából lényegtelen, például a teljesen értelmetlen job-control programban elkövetett helyesírási hibákat is könyörtelenül büntették. Így a programozókban olyan érzés alakult ki, hogy a programokat a gép és az operációs rendszer ellen küzdve és nem azzal együttműködve kell kifejleszteni.

Nem véletlen, hogy ebben az időben alakult ki és terjedt el a strukturált programozás. E. W. Dijkstra, C. A. P. Hoare és a többiek a programozáselmélet alapjainak erősítésében keresték a megoldást. Ugyancsak ez idő tájt láttak napvilágot a szoftvertermékek iparszerű előállítására vonatkozó, ma már jórészt elfelejtett elméletek is.

A megújulás első kézzelfogható jelei azonban csak 1972-ben az Intel 8080 típusú, 8 bites mikroprocesszor, valamint a B. W. Kernighan és D. M. Ritchie által kidolgozott C nyelv képében jelentkeztek. A 8080-as mai léptékkel mért szerény teljesítménye a hajlékony lemeztárral együtt már lehetővé tette egy barátságos, interaktív operációs rendszer, a CP/M létrehozását. A szerző G. Kildall volt, az időpont 1974 nyara.

1975-től az események felgyorsulnak. Megjelenik a Zilog Z80-as, a 8080-nal felülről kompatibilis, de még 8 bites processzora. A 8 bites mikrogépek alapvető programozói nyelve egyelőre az 1964-ben született, tehát már több mint 10 éves BASIC, melynek bölcsőjét a többi között a magyar származású Kemény János ringatta.

A BASIC nyelv sikere jó bizonyítéka annak, hogy a felhasználók megcsömörlöttek a túlzottan tudományos (ALGOL—68), valamint a monumentális hardvert igénylő (PL/I) barátságtalan nyelvektől. Olyan nyelvet akartak, amit könnyen megértenek, és amellyel azonnal sikerélményük lehet egy 8 vagy 16 kbájtos gépen. A BASIC ilyen célokra — minden elvi és gyakorlati korlátja ellenére — kiválónak bizonyult.

Amikor azután a 8 bites processzorok ára nagymértékben csökkent (a MOS Technology 6502 például 1975 nyarán 25 dollárba került, szemben a 8080-as mintegy 150 dolláros árával), a személyi számítógépek szaporodása minden eddigi rekordot megdöntött. Az IBM sem akart kimaradni az új piacról, ezért 1975 szeptemberében kiadta az 5100-at, az első táskaméretű személyi gépet (16 k RAM, BASIC, mágneskazettás háttér), kb. 9000 dollárért.

A világon jelenleg is a legismertebb mikrogépes folyóirat, a BYTE első száma 1975 szeptemberében jelent meg. A 16 bites processzorok megjelenésével (az elsőt a Texas Instruments hozta létre TMS9000 néven 1976-ban) megnyílt az út a professzionális személyi számítógépek előtt.

1981-ben az IBM PC-vel együtt a Microsoft új operációs rendszere, az MS—DOS is a piacra került. Ezek a gépek elég nagy teljesítményűek voltak, és az 1980 óta forgalomban lévő Winchester típusú mágneslemezekkel már alkalmasak lehettek komolyabb operációs rendszerek üzemeltetésére is. Ekkor hódított igazán az eredetileg DEC minigépeken fejlesztett UNIX, és megjelent a San Diego-i Egyetem UCSD rendszere, egy Pascal alapú, gépfüggetlen operációs rendszer.

Összefoglalva: a 80-as évek elejére a számítástechnikában új irányzat lett úrrá, amely mást és sokkal többet ígért a programozóknak, mint a nagygépek korszaka. A fejlődés spirális ciklust járt be, a személyi számítógépekkel egy magasabb szinten tértünk vissza az első és második generációs gépek emberközeli felhasználásához. A régi gépeket alacsony teljesítményük miatt lehetett csak „személyesen" használni, a mikrogépeket pedig alacsony áruk miatt hozhatjuk emberközelbe.

A jelen

Úgy tűnt, hogy a professzionális vonalon az IBM PC (1981), a szórakoztató elektronika vonalán pedig a Commodore 64 (1982) megjelenésével a mikroszámítógépek már döntő szerephez jutottak a számítástechnikában. A kialakult helyzet több szempontból is figyelemreméltó. Mi elsősorban a szakma — a programozás — szempontjából adunk elemzést, de néhány mondatban ki kell térnünk a mikroelektronika társadalmi szerepére is.

A számítástechnika évtizedekig meglehetősen titokzatos, csak professzionalista módon művelhető szakma, illetve tudomány volt. A személyi számítógépek terjedésével — és itt elsősorban a Commodore 64, a Sinclair és az Atari gépekre gondolunk — megindult a számítástechnika társadalmasodása. Ezek a gépek sok, a számítástechnikában nem jártas embernél olyan használati tárgyakká váltak, mint a televízió, a mosógép vagy a videomagnó. Az emberek megszokják, hogy eggyel több olyan eszközük van, melynek működését ugyan csak felületesen ismerik, de sok ügyes dologra képes — a sakkozástól az írógép magasabb szinten való helyettesítéséig. Ez a társadalmi hatás természetesen szorosan összefügg a szakmában már lezajlott, illetve most is folyó változásokkal.

Ezzel kapcsolatban elsősorban az általá-

nos szakmai stratégiával, a piacon levő szoftvertermékekkel, valamint e termékek előállításának eszközeivel célszerű foglalkozni.

Az átlagos felhasználó, aki úgy használ személyi számítógépet, mint egy hi-fi berendezést vagy egy videomagnót, nyilvánvaló módon kész szoftvert igényel és kap is. Ezek a termékek — számítástechnikai szóhasználattal: célrendszerek — négy alapvető felhasználási területre készültek: szövegfeldolgozásra, táblázatkalkulátornak, adatbázis-kezelésre és játék céljára.

Mindegyik kategóriában kialakultak a sztárok, ezek utánzatai és az újat ígérő trónkövetelők is.

A szövegfeldolgozás területén az áttörést a WordStar hozta, 1979-ben. Ez volt az első széles körben elterjedt professzionális szövegkezelő rendszer, melynek karrierje ma is tart. Eredetileg 8 bites gépekre készült, de jelenleg már minden számottevő mikrogépen megtalálható.

Az utóbbi 3-4 évben megjelent szövegfeldolgozó rendszerek: Easy Writer, Easy Script, Script, Professional Editor, Personal Editor stb. lényegében újítást nem tartalmaznak, általában csak egy-egy kis ötlettel, a szolgáltatások ésszerű csoportosításával, a felhasználóval való kapcsolat minél barátságosabbá tételével igyekeznek kiemelkedni a tömegből.

A táblázatkalkulátorok (spread-sheet) „WordStar"-ja a vele egyidős VisiCalc, amely ma is az élvonalban van. E rendszerek lényege, hogy egy kétdimenziós táblázat elemeire azok koordinátáival hivatkozhatunk, az elemek értékei pedig nemcsak számok vagy szövegek, hanem formulák is lehetnek, melyek a táblázaton belüli összefüggéseket fejeznek ki. A táblázat bármely elemének módosításakor a rendszer az összefüggések alapján az érintett elemeket is módosítja. Meg kell jegyezni, hogy a táblázatkalkulátorok területén a VisiCalc hegemóniája erősebb, mint a WordStaré a szövegfeldolgozásban. Mellette csupán néhány rendszer, mint a Multiplan és a CalcResult jutott igazán szóhoz. Ugyanakkor olyan véleményeket is hallani, hogy a táblázatkalkulátorok Európában kevésbé népszerűek, mint az Egyesült Államokban.

Az első két célrendszer-családnál nagyobb fajsúlyú termékeknek tartják az adatbázis-kezelő rendszereket. Ennek oka azonban részben történelmi, mivel a nagygépes adatkezelés már tekintélyes múlttal rendelkezik, részben pedig azt a tényt tükrözi, hogy az adatbázis-kezelő rendszerek ténylegesen a legbonyolultabb mikrogépes szoftvertermékek közé tartoznak.

Az első mikrogépes rendszereket a nagygépesek még egyszerű fájlkezelőknek nevezték, de ez a helyzet már 1979-ben megváltozott. Ekkor jelent meg Wayne Ratliff relációs adatbázis-kezelő rendszere, a Vulcan, melyet később az Ashton-Tate cég dBASE II néven forgalmazott. Ez a rendszer a 8 bites gépeken világszerte elterjedt, és egyértelműen bizonyította: a mikrogépek igen sok adatkezelési feladat megoldásánál nemcsak használhatók, hanem a legmodernebb elvű adatbázis-kezelési technikák megvalósítására is képesek. A mikrogépek viszonylag korlátozott tárkapacitásáért és sebességéért sok esetben a rendszer „személyes varázsa", a barátságos felhasználói kapcsolat kárpótol.

Bár a dBASE II a legtöbb 8 bites és utódja a dBASE III a legtöbb 16 bites gépen elterjedt, mégsem nevezhetők egyeduralkodónak az adatbázis-kezelés területén. Jóformán minden jobb szoftverház létrehozta a saját rendszerét, és a piacot a szó szoros értelmében elárasztották a lényegében egyformán jó rendszerek. Mi csak ízelítőül sorolunk fel néhányat: CONDOR, VisiFill, R:base 400, Infostar, Knowledge Man.

Ezek mindegyike IBM PC-n működik, és mintegy 500 dollárba kerül. Érdekességként említhető, hogy olyan hobbigépeken is, mint a Commodore 64, igen jól használható rendszerek léteznek: SUPERBASE, MULTIDATA.

Komoly felhasználók a játékprogramokat általában méltóságukon aluli témaként kezelik, pedig ezek a termékek több szempontból is figyelemreméltóak: egyrészt olyan tömegben jelennek meg a piacon, hogy nem lehet figyelmen kívül hagyni őket, másrészt mind a jó logikai, mind az ötletes grafikát felvonultató játékok mögött nem lebecsülendő szakértői, rendszerterveői és programozói teljesítmény van. Ha valaki megnézi a könyv alakban is kiadott Sargon sakkozóprogram szerkezetét, meggyőződhet arról, hogy egy operációs rendszer bonyolultsága szintjén lévő rendszerről van szó.

Integrált rendszerek

A fejlődés természetesen nem áll meg, és a következő kézenfekvő lépcsőfok a célrendszerek területén az ún. integrált rendszerek megjelenése. Az előfutár 1982-ben a Lotus Development Corporation terméke, az 1-2-3 volt, amely a táblázatkalkulátort grafikával és adatkezeléssel kombinálja. Az integrált rendszerek vonzereje abban van, hogy több alkalmazási területet egy rendszerrel fednek le, és így a felhasználó egymással jól kommunikáló adatbázis-kezelő, szövegfeldolgozó és egyéb rendszereket kap egy csomagban.

Mivel az integrált rendszerek hardverigénye meghaladja a korábbi célrendszerekét, ezek már csak professzionális mikrogépeken futnak, és ez elsősorban az IBM PC különböző változatait jelenti.

A Lotus 1984-ben jött ki Symphony nevű új integrált rendszerével, amely már „mindent" tartalmaz, ami az 1-2-3-ból kimaradt. Sikerét valószínűleg komplexitása is limitálta. A versenytársak közül talán a Framework (Ashton-Tate 1984) és a Golden Gate (Cullinet 1984) a legsikeresebbek, de egyedül az IBM idei programkatalógusában 11 integrált rendszer szerepel.

Ha egy kicsit mélyebbre nézünk a professzionális mikrogépek szoftverrendszerébe, szemügyre vehetjük azokat az operációs rendszereket, amelyek alatt az alkalmazói rendszerek működnek. A domináns megint az IBM, a többiek — néhány érdekes kivételtől eltekintve — ők utánozzák.

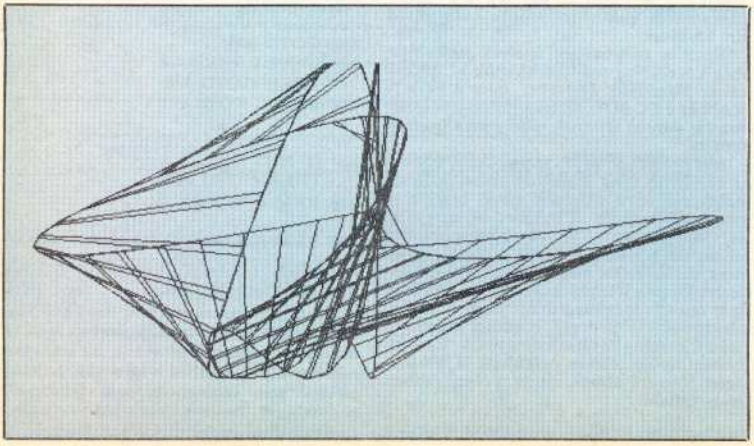
Az IBM-nél a PC DOS az uralkodó rendszer, de mindenképpen említést érdemel az UCSD P—System is, melyet Pascal-alapokon a San Diego-i egyetemen fejlesztettek. A többfelhasználós (IBM PC AT) időosztásos rendszereken az UNIX egyik mikrogépes változata, a PC XERNIX használható a PC DOS 3.1, a lokális hálózatot támogató rendszer mellett.

A versenytársak az IBM-et olyan területen akarták legyőzni, ahol az kevésbé járatos, és ezért hozták létre az újszerű, integrált operációs rendszereket.

Ezek a rendszerek a laikus felhasználókat megszabadították a programozástól azzal, hogy az alaprendszerbe beépítették a fontosabb alkalmazási célrendszereket (adatbázis-kezelés, szövegfeldolgozás, kalkulátor stb.), sőt a felhasználónak sok esetben írnia sem kell, mert a rendszert nem parancsokkal, hanem egy képekből, ún. ikonokból álló menü elemeire való mutogatással vezérli.

Az első széles körben elterjedt integrált operációs rendszer az Apple cég által 1983-ban piacra dobott Lisa volt, amely természetesen új hardverkonfigurációt is jelentett. Ez a gép sok új megoldást hozott, de különösen mai szemmel, meglehetősen lassú volt.

Az Apple következő új gépe jóformán pontosan egy évvel a Lisa után jelent meg, Macintosh néven. A Macintosh különösen az irodákban vált népszerűvé, hiszen prog-



ramozás nélkül a legtöbb irodai tevékenység támogatására alkalmas. Ára a Lisa eredeti árának felénél alig volt magasabb.

Az integrált operációs rendszerek versenyéből a Commodore sem akar kimaradni, és 1985 augusztusában bejelentette új gépét az AMIGA-t. A gép egy Motorola 6800 központi processzorra épül, melynek órafrekvenciája 7,1 MHz.

Az elődöknél tapasztalható viszonylag lassú működést az AMIGA speciális céláramkörök beépítésével igyekszik kiküszöbölni. Így a 68 000-et több tevékenység alól mentesítik, az animációt, a grafikát, a perifériakezelést külön erre a célra tervezett integrált áramkörök, ún. custon chipek valósítják meg. A gép ára természetesen a Macintosh árának mintegy a fele lesz, 1300 dollár.

Programozási nyelvek

Bár az integrált operációs rendszerű gépek a felhasználót nem bátorítják programozásra, mégis érdemes megvizsgálni a mikrogépeken használatos nyelveket.

A „nép nyelve", amit a legtöbben használnak, szidnak és dicsérnek, a több mint 20 évvel ezelőtt magyar közreműködéssel létrejött BASIC. A professzionalista programozási elveket sokszor figyelmen kívül hagyó, az interpretatív megvalósítás következtében elég lassú nyelvnek van egy olyan érdeme, amiért mindent meg kell bocsátani neki: soha nem látott tömegek „tanultak meg" programozni, mivel a nyelv egyszerű és barátságos.

Mivel a vevők kiszolgálása igen aprólékos, ugyanakkor viszonylag nagy adattömeget mozgató programok írását követelte, az alkalmazói rendszerek sebessége szűk keresztmetszetnek bizonyult. Ez elsősorban a 8, majd csökkenő mértékben a 16 és 32 bites processzorokra alapozott gépeknél jelent problémát.

Az egyik, az általánosabb megoldás az volt, hogy visszatértek az első és második generációs gépeknél jobb híján használt gépi kódhoz, illetve az assembly nyelven való programozáshoz, természetesen a korábbinál jóval magasabb szinten. Az assemblerek újra előtérbe kerültek, és a legtöbb sikeres célrendszert, például a WordStart és a VisiCalcot ebben írták.

A mai assembleren alapuló fejlesztő rendszereket természetesen nem szabad az 50-es és 60-as évek fordulóján használt alacsony szintű programozási eszközökhöz hasonlítani. Ezek általában ún. keresztfejlesztést támogatnak, ahol cserélhető kódgenerátorokkal egy nagyobb teljesítményű fejlesztőgépen többféle célgépre lehet tárgykódot előállítani. Így a fejlesztés hatékonysága erősen megnő, és a szoftver fejlesztése a hardvertervezéssel párhuzamosítható, hiszen a konkrét működő gépre a fejlesztés idején még nincs szükség.

A hatékony rendszerek előállításának másik útja az volt, hogy igyekeztek olyan nyelveket bevezetni, amelyek egyesítik az assembly nyelv hatékonyságát a magas szintű nyelvek kényelmével.

Az egyik ilyen kompromisszum eredménye volt a már meglévő FORTH nyelv mikrogépeken való bevezetése, amely nagyobb mértékben 1979-ben indult el, elsősorban a 8 bites gépeken. A FORTH olyan nyitott nyelv, amelyben egy fejlesztő laboratórium viszonylag könnyen építhet ki saját céljainak legjobban megfelelő programozási környezetet. A nyelv a strukturált programozást messzemenően támogatja.

A kényelmes, modern elveken nyugvó programozás támogatására, amikor már nem a tárgykód futási sebessége a fő szempont, a mikrogépeken is megjelentek a népszerű, magas szintű nyelvek is. Itt elsősorban a Pascalt és a C-t kell említeni. A Pascal legismertebb változata az UCSD Pascal, melyet az operációs rendszereknél is említettünk. A C alapvetően az UNIX nyelve, de a mikrogépeken ettől függetlenül is elterjedt. Ez talán az az eredetileg rendszerprogramozásra szánt nyelv, melynek népszerűsége születése, 1974 óta állandóan növekszik. A nyelv szerzői nemcsak kiváló szakemberek, hanem látnoki képességekkel is rendelkeznek.

A mikrogépek varázsa természetesen az oktatást sem hagyta érintetlenül. A számos különböző célú oktatórendszer közül kiemelkedik egy nyelv, a LOGO, amely 1982 óta egyre növekvő népszerűségnek örvend. A programozás alapvető diszciplináit (eljárások, rekurzió, listakezelés) a LOGO kiválóan egyesíti a kezdőket és a gyerekeket megnyerő, népszerű eszközökkel, mint a színes grafika, a barátságos felhasználói kapcsolat, és ezért a nyelv a programozás jóformán bármilyen szinten való oktatására kiválóan alkalmas.

Érdekes módon az új elvű, logikai programozás is megjelent a PROLOG nyelv személyében. Ezen a nyelven a programok tulajdonképpen következtetési rendszerek formájában jelennek meg, ami bizonyos feladatok esetén a fejlesztőknek nagy támogatást jelent. A szóban forgó feladatok nagy része a mesterséges intelligencia körébe tartozik. A megfelelő alkalmazási célrendszerek, az ún. szakértői rendszerek feladata, hogy valamely speciális szakterület művelőit, a szakértőket úgy vonja be a számítástechnikát alkalmazók körébe, hogy azoknak különösebb számítástechnikai ismeretet ne kelljen elsajátítaniuk, hanem minden energiájukat saját szakmai problémáik megoldására fordíthassák.

A szakértői rendszerek jelenlegi fő területei az orvostudomány, a geológia, az építészet és a biofizika, de egyéb humán tudományokban való elterjedésük csak rövid idő kérdése.

Ezzel a mikrogépes nyelvek arzenálját még közelítőleg sem merithettük ki, hiszen olyan klasszikus nyelveket, mint a FORTRAN és a COBOL, melyek mikrogépes változatai szintén léteznek, eddig nem is említettünk.

Mivel cikkünk egyes alcímei egy, a jövőt is felvázoló részt sugallnak, most valószínűleg csalódást okozunk azzal, hogy erre nem vállalkozunk. A fejlődés olyan gyors, és Magyarország ennek középpontjától olyan messze esik, hogy itt prófétái ugyan nem, de apostolai még lehetünk a számítástechnika eme irányzatának. Annyi azonban már most is megállapítható, hogy a mikroelektronika talán legfőbb eddigi érdeme a számítástechnika társadalmasítása. Ez a folyamat természetesen nélkülünk is lezajlik, ezért célszerű abban minél tevékenyebben részt vállalni.

BAKOS TAMÁS

Strukturá

A strukturált programozásnak több megközelítése és meghatározása lehet. Mi a továbbiakban tekintsük kiindulásnak azt a megközelítést, amely szerint a strukturált programozás azt jelenti, hogy programjaink felépítésénél csak háromfajta vezérlőszerkezetet használunk: a szekvenciákat, szelekciókat és iterációkat. A vezérlőszerkezetek azt határozzák meg, hogy a program különböző részei időben hogyan követik egymást, illetve hogy egy adott időpontban mely részek hajtódnak végre.

Egy A program vagy programrész a B, C és D részek szekvenciája, ha az A végrehajtása B, C és D egymás utáni végrehajtását jelenti ebben a sorrendben. A szekvencia diagramon való ábrázolását Jackson nyomán az *1. ábra* mutatja.

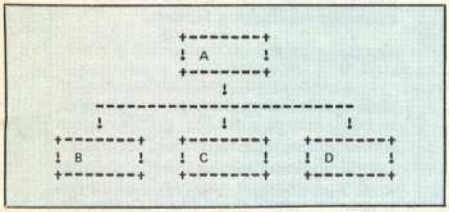
A szekvencia megvalósítása valamely programnyelven úgy történik, hogy a megfelelő részeket vezérlési utasítások nélkül, egymás után írjuk.

Egy A program vagy programrész a B, C és D részek szelekciója, ha A végrehajtása vagy B vagy C vagy D végrehajtását jelenti. A „vagy" szót itt kizáró értelemben használtuk, azaz az alternatívák közül egyet és csak egyet kell végrehajtani. A *2. ábra* a szelekció diagram formában való ábrázolását tartalmazza. Azt, hogy az alternatívák közül melyiket kell végrehajtani, logikai feltételek határozzák meg. Az ábrán látható szelekció megvalósítása BASIC nyelven (sorozatunkban a továbbiakban a mikroszámítógépeken leginkább elterjedt BASIC nyelvet fogjuk hivatkozási nyelvként használni, jóllehet látni fogjuk, hogy a BASIC vezérlési szerkezetei nem mindig a legkézenfekvőbb szerkezetek a strukturált programozás szempontjából):

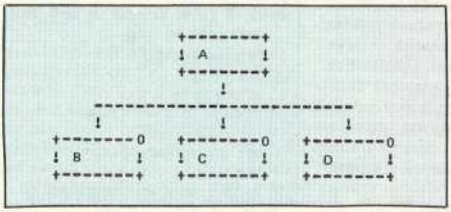
```
100 IF NOT feltétel—1
    THEN GOTO 200
110 REM B KÓDJA
    :
    :
190 GOTO 400
200 IF NOT feltétel—2
    THEN GOTO 300
210 REM C KÓDJA
    :
    :
290 GOTO 400
300 IF NOT feltétel—3
    THEN GOTO 400
310 REM D KÓDJA
    :
    :
400 REM A SZELEKCIO UTANI RESZ
```

A feltétel—1 teljesülése esetén a B részt, feltétel—2 teljesülése esetén a C részt, feltétel—3 teljesülése esetén a D részt kell végrehajtani. A 300-as címkéjű sorban lévő IF utasítás el is maradhat akkor, ha a C részt minden olyan esetben végre kell hajtani, amikor sem feltétel—1, sem feltétel—2 nem teljesül. Természetesen 300-as címkéjű sornak ebben az esetben is kell léteznie.

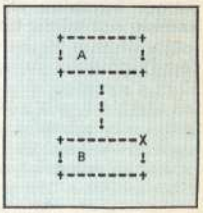
programtervezés



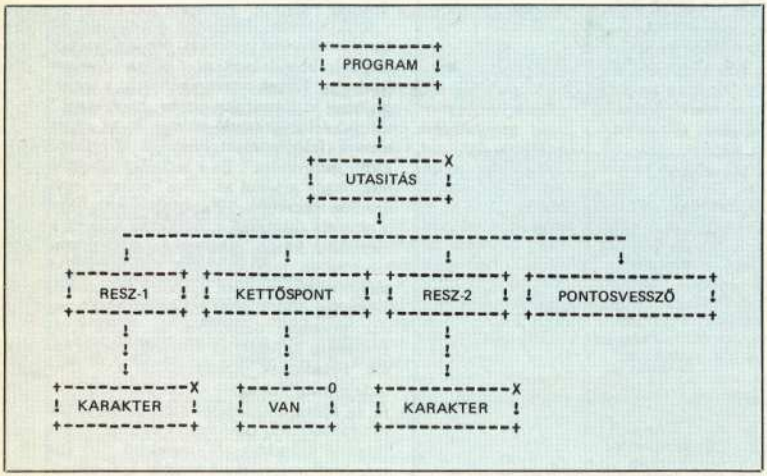
1. ábra



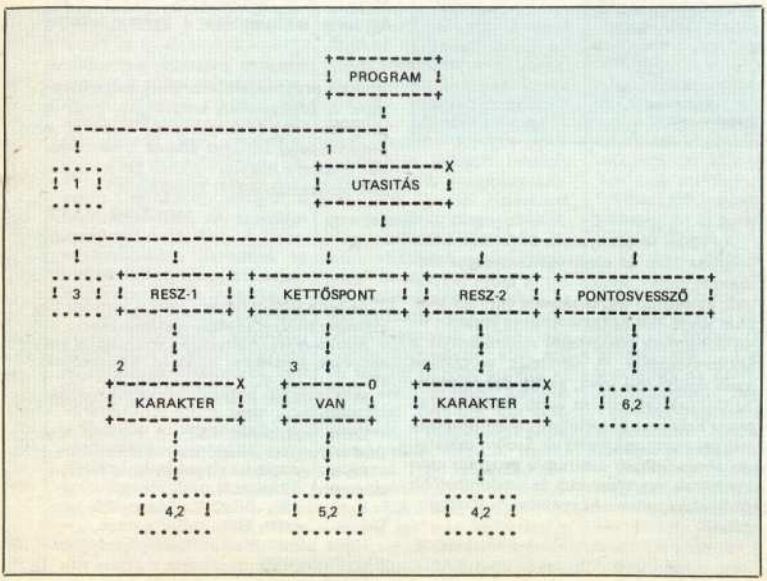
2. ábra



3. ábra



4. ábra



5. ábra

Első pillanatban talán meglepőnek tűnhet, hogy a fenti programrészben öt GOTO utasítás szerepel, bár a strukturált programozást GOTO nélküli programozásnak is szokták nevezni. Az itt szereplő GOTO utasítások azonban „ellenőrizhető" GOTO utasítások abban az értelemben, hogy feladatuk csupán a szelekció megvalósítása. Ezért az olyan nyelvekben, amelyekben egymásba ágyazható IF... THEN... ELSE... utasítások léteznek (például COBOL, PL/I, PASCAL, C nyelv), nincs is szükség rájuk. Amint a későbbiekben látni fogjuk, a programtervek kódolása esetén ezeknek a GOTO utasításoknak az alkalmazása mechanikusan történik.

Egy A program vagy programrész a B rész iterációja, ha A végrehajtása B egymás utáni végrehajtásait jelenti. Mielőtt a B rész végrehajtódna, egy feltétel kiértékelődik, és a végrehajtásra csak akkor kerül sor, ha a feltétel igaz („while" feltétel). Az is előfordulhat, hogy a B rész egyszer sem hajtódik végre. Az iteráció ábrázolását diagramokon a *3. ábrán* láthatjuk. A fentiekből látható, hogy az iteráció tulajdonképpen csak a ciklusnak egy más elnevezése.

Ha az iterációban induláskor egy változó egy adott kezdőértéket kap, ha ez a változó a mag minden egyes végrehajtásakor ugyanazzal az értékkel növekszik vagy csökken, és ha az iterációt addig kell végrehajtani, amig ez a változó egy adott értéket el nem ér, a FOR-NEXT utasításpár használható az iteráció megvalósítására. Egyébként az iteráció a következő módon kódolható:

```
100 IF NOT feltétel
    THEN GOTO 200
110 REM B KÓDJA
    :
    :
190 GOTO 100
200 REM AZ ITERACIO UTAN
    KOVETKEZO KOD
```

A strukturált programozás alapfogalmainak ismertetése után a strukturált programtervezésről lesz szó. Valamilyen értelemben vett programtervezést minden programozó végez, hiszen mielőtt a kód irásához kezdene, először átgondolja a megoldandó feladatot, és legalább nagy vonalakban próbálja még a kód írása előtt meghatározni a

megoldás útját. Ez a folyamat azonban túl hamar befejeződik, és egy homályos elképzelés után már a kód írása következik. A rendelkezésre álló nyelvek, vagy hagyományos értelemben vett blokkdiagramok azonban nem mindig bizonyulnak kellően magas szintű és tömör eszköznek a tervezéssel kapcsolatos gondolatok kijelzéséhez. Ezért a strukturált programtervezés Jackson által kidolgozott módszerének alkalmazása esetén először a program vezérlési szerkezete határozódik meg, és erre a szerkezetre, mint vázra helyezkednek el a program utasításai.

A továbbiakban illusztrációként a következő feladat megoldását fogjuk bemutatni: Az a$ tömb egy olyan szöveget tartalmaz, amely kisebb egységekből, „utasításokból" áll. Ezeket az utasításokat pontosvessző zárja le. Készítsük el azt a programot, amely a szöveget a b$ tömbbe másolja, és az utasításokban esetleg előforduló, első, kettősponttal lezárt részeket („címke") ezek sorszámaival helyettesíti.

A tervezés első lépéseként meghatározzuk a program szerkezetét. Ez azt jelenti, hogy meghatározzuk, hogy a program milyen egymásba ágyazott vezérlési szerkezetekből épül fel. A vezérlési szerkezetek ábrázolására az előzőekben ismertetett diagramokat használjuk. A teljes programszerkezetet tartalmazó diagram a *4. ábrán* található.

A szerkezet meghatározása után következik a további szerkezettel nem rendelkező, ún. elemi tevékenységek és feltételek elhelyezése a szerkezeti ábrán. Ennek érdekében sorszámokkal ellátva felsoroljuk a program folyamán használt tevékenységeket és feltételeket. Ezek BASIC utasítások, illetve BASIC nyelven megadott logikai feltételek. A fent felsorolt és a következő lépések nem választhatók el egymástól mereven, hiszen például a szerkezet meghatározásakor már van valami elképzelésünk az elemi tevékenységekről és feltételekről is, de ebben a lépésben az elsődleges feladat mégis a szerkezet meghatározása. Az is előfordulhat, hogy egy későbbi lépésben felmerülő probléma megoldása szükségessé teszi egy előző lépés eredményének a megváltoztatását. A tervezés tehát iteratív folyamat.

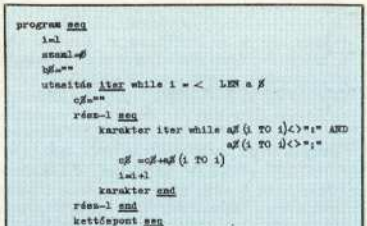
A feladat megoldása folyamán a ZX-Spectrum BASIC nyelvét használjuk. Az elemi tevékenységek az alábbiak:

```
1 LET i=1: LET szaml=0:
  LET b$=""
2 LET i=i+1
3 LET c$=""
4 LET c$=c$+a$(i TO i)
5 LET szaml=szaml+1
  LET c$=STR$(szaml)+":"
6 LET b$=b$+c$+":"
A feltételek:
1 i=<LEN a$
2 a$(i TO i)<>":"
  AND a$(i TO i)<>":"
```

```
10 LET i=1: LET szaml=0:
   LET b$=""
20 IF i<=LEN a$ THEN GOTO 100
30 LET c$=""
40 IF a$ (i TO i) =":" OR a$ (i TO i)=";"
   THEN GOTO 60
50 LET c$=c$+a$ (i TO i) :
   LET i=i+1:
   GOTO 40
60 IF a$ (i TO i) =":" THEN
   LET szaml=szaml+1:
   LET c$=STR$(szaml) +":"
   LET i=i+1
70 IF a$ i TO i =';' THEN GOTO 90
80 LET c$=c$+a$ (i TO i) :
   LET i=i+1: GOTO 70
90 LET b$=b$+c$+";":
   LET i=i+1:
   GOTO 20
100
        :
```

6. ábra

7. ábra



3 a$ (i TO i)=":"
4 a$ (i TO i)<>":"

Az elemi tevékenységek és feltételek felsorolása után az elemi tevékenységek sorszámait azon „doboz" elé és mögé helyezzük, amelynek végrehajtása előtt vagy után őket végre kell hajtani. Az ábra legalján lévő dobozokra közvetlenül ráilleszthetjük a tevékenységeket. A feltételek sorszámait azon doboz fölé írjuk, amely doboz végrehajtásának feltétele az adott sorszámú feltétel. Feladatunk esetében az így kiegészített programszerkezetet az *5. ábra* mutatja. Az ábrán látható diagram a program szerkezetének egy tömörebb és áttekinthetőbb ábrázolása, mint a hagyományos blokkdiagram.

A program elkészítésének következő lépése a szerkezeti ábra lekódolása BASIC nyelven. Feladatunk BASIC kódját a *6. ábra* tartalmazza. Ez a lépés mechanikusan elvégezhető. Esetünkben a feltételek tagadására nem a NOT logikai műveletet használtuk, hanem a De-Morgan formulák alkalmazásával átírtuk a feltételt.

Megjegyzések

1. A „kettőspont" rész kódolásánál eltértünk az előzőekben ismertetett módszertől, és felhasználtuk a BASIC által biztosított lehetőséget egyágú szelekciók kódolására.

2. A szerkezet meghatározásánál éltünk azzal a lehetőséggel, hogy egy iteráció üres is lehet. A „rész—2" iteráció minden olyan esetben üres, amikor nincs kettőspont az utasításban.

3. A tervezés folyamán használtuk az ún. előreolvasási technikát; esetünkben egy karakter olvasásának az *i* index növelése felel meg. Ennek a technikának az a lényege, hogy az algoritmus elején „beolvassuk" az első feldolgozandó elemet, és az egyes elemek feldolgozását követi a következő elem „beolvasása". Ez a technika lehetővé teszi, hogy például az „üres" eseteket egyszerűen lekezeljük. Megemlítjük még, hogy a tervezés folyamán a szerkezeti ábra és a forráskód között átmenetként alkalmazni lehet az ún. pszeudokódot. A pszeudokód tulajdonképpen olyan nyelvnek tekinthető, amelyben csak a vezérlési szerkezetek vannak pontosan definiálva, az elemi tevékenységek és feltételek akárhogy megadhatók. Feladatunk pszeudokódját a *7. ábra* tartalmazza. Az egyes szerkezetek jelölésére az ábrán aláhúzással kiemelt kulcsszók szolgálnak: seq (szekvencia számára), iter (iteráció számára), sel (szelekció első ága számára), or (szelekció további ágai számára). Az egyes szerkezeteket az end kulcsszó zárja le. A pszeudokód írása közben a beágyazott szerkezeteket a sorban beljebb kezdjük.

Amint láthattuk, a tervezés legfontosabb lépése a program szerkezetének meghatározása. A folytatásban ismertetjük, hogy a program által feldolgozandó, valamint a program által készített adatok szerkezeteinek elemzése hogyan segítheti elő a megfelelő programszerkezet kialakítását.

MOLNÁR MÁTÉ

(folytatjuk)

Akiket a téma bővebben érdekel, azoknak figyelmébe ajánljuk az alábbi szakirodalmat:

Aszalós János: A strukturált programozás irodalmának áttekintése. SZÁMKI Közlemények 19.

M. A. Jackson: Principles of Program Design. Academic Press, 1976.

Leifer Ingevalsson: J.S.P.: A Practical Method of Program Design. Student Litteratur.

Dr. Mérey András: Programtervezés Jackson-módszerrel. SZÁMALK, 1983.

Hernetzky K.—Orbán K.: Jackson-féle tervezési módszer. TII, 1982.

Urbán János: Matematikai logika. Műszaki Könyvkiadó, 1983.

Akik csatát veszíthetnek, de háborút nem

Két előző számunkban megismerkedtünk a mikroszámítógépes forradalom első szakaszának három nagy vállalkozó egyéniségével, Clive Sinclairrel, Jack Tramiellel és Steve Jobsszal. Eddigi elbeszélésünk sikereik fénykorában mutatták be ezeket a nem kevés egyéni önbizalommal rendelkező embereket. Ma már mindhárman a perifériára szorultak, és igen kevés esélyük van arra, hogy a felhasználók a továbbiakban is többre fogják értékelni nagy felhajtással kísért akcióikat, mint a megalapozott üzleti vállalkozások stabilizálódó kínálatát. Mi is történt tulajdonképpen?

Kockázatos üzletpolitika

Az eladási darabszámok növelésének bűvöletében mind Clive Sinclair, mind Jack Tramiel, a tömegcikk-kereskedelmi (mass merchandising), csomagküldő és áruházi/általános kiskereskedelmi üzlethálózati értékesítési formákra helyezték a hangsúlyt. Ez kiválóan működött az elmúlt évek impulzív tömegvásárlásainak időszakában, amikor széles rétegek érdeklődésének homlokterébe került az új elektronikus technika. Az embereknek nem sok fogalmuk volt a számítógépekről, a felkínált áruban csak újabb elektronikus kacatot láttak, amiről úgy gondolták, hogy nem árt kipróbálni. Ekkor nyilvánvalóan azt az árut vették leginkább, amelyik a legolcsóbb volt, de kinézetre azért szolidnak tűnt. A Commodore eddigi sikereinek ez utóbbi volt az alapja. Sinclair sikerformulája az amerikaiaknál kevésbé tehetős országok vásárló tömegeihez igazodott, nevezetesen a „minél nagyobb teljesítményű elektronika, minél olcsóbb köntösben" elvet követte.

Ezzel az üzletpolitikával elkülönítették magukat az egyre növekvő piaci befolyást szerző számítógépes szaküzletektől. Ez alapvető stratégiai hiba volt, mert a fejlett tőkés országok mai vásárlói már tudatosan veszik a számítógépet, ehhez pedig a megfelelő kereskedelmi támogatást csak a számítógépes szaküzletekben kaphatják meg. A vevőket az olcsó ár is kevésbé vonzza már. Sokkal fontosabb, hogy a kereskedő meg tudja győzni őket a vásárolandó gép többirányú hasznosságáról. Megfelelő meggyőzés esetén akár a korábbit jóval meghaladó összegeket is hajlandók kiadni erre a célra.

Clive Sinclair ráadásul túlzásba vitte találmánycentrikusságát is. Lapos zsebtévé, elektromos autó — ezek olyan, önmagukban is rengeteg piaci kockázatot hordozó termékek, amelyek aligha növelhették magánvállalatának, a Sinclair Research-nek az üzleti biztonságát. 1985-ben, ha nem is jutott csődbe, de bizonyos pénzügyi nehézségekkel került szembe a Sinclair Research. A tömegáru PC-piac drasztikus visszaesése is igen nagy mértékben hozzájárult ehhez. A komoly alkalmazási szoftverrel együtt árult, korszerű Sinclair QL sem menthette meg a céget, mivel egy ilyen gépet már igen nehéz a szaküzlethálózaton kívüli csatornákon sikerrel értékesíteni.

A részvényesek ellenőrzése alatt álló Commodore bizonyos belső erői már Tramiel 1983. évi nagy piaci akciója során rádöbbentek, hogy micsoda öngyilkos politikát folytat főnökük. Ahogy az amerikaiak mondják: „Tramiel kannibálszerű módon kívánta elfogyasztani a piacot". Intenzív belső akciók eredményeképpen Jack Tramiel hatáskörét igen jelentősen megnyirbálták. Nem váltották ugyan le, de mint vállalkozó, „parkolópályára" került.

Az események egyébként a részvényeseket és a józan belső erőket igazolták. 1984 még Commodore-sikerév volt, mivel tovább sikerült növelni az 1983. évi 1 milliárd dolláros forgalmat. 1985-ben jött a feketeleves. A forgalom az előző évinek több mint a felével csökkent. Több százmillió dolláros veszteséget kellett elviselnie a vállalatnak, és új Amiga gépével ma már kénytelen a fennmaradásért küzdelmet folytatni.

Tramiel feltámadása

Tramiel 1983—84 fordulóján „béklyóba kötve" találta magát. Egyre rosszabb szájízzel járt be munkahelyére. Ha meggondoljuk, ez igen furcsa dolog egy nagyvállalat elnökének esetében. 1984 februárjában nem volt képes továbbra is elviselni helyzetét, és az egyetlen lehetséges utat választotta: kilépett a cégtől. Először még pihenni ment feleségével, de hamarosan rá kellett döbbennie, hogy munka nélkül nem élet az ő élete. Fiaival magáncéget alapított, hogy elölről kezdjen mindent. Ez sem elégíthette azonban ki nagyvállalkozói ambícióit.

1984. július 2-án robbant a bomba. Tramiel felvásárolta anyavállalatától, a Warnertől az addigra tulajdonképpen a csőd szélére került Atarit. A Warnernek gyakorlatilag ingyen kellett odaadnia cégét. Tramiel nem fizetett készpénzt, hanem egy olyan kötelezvényt írt alá, hogy 10 év múlva 140 millió dollárt fizet a Warnernek 13% kamat mellett, illetve állandóan fizetendő, 9%-os kamat mellett 100 millió dollárt 12 év múlva. Ebben a vállalásban egy önmagát megnevezni nem kívánó tőkéscsoport is támogatta Tramielt. A Warnernek ráadásul 425 millió dolláros veszteséget kellett magára vállalnia a vétel pillanatában. Mindennek fejében csak azt kapta, hogy egy későbbi időpontban megvásárolhatja az új vállalat részvényeinek 32%-át.

Az üzlet különös iróniája, hogy pontosan Tramiel korábbi, „piaclenyelési" akciója juttatta erre a sorsra az Atarit. Érdekes volt hallani Tramiel első nyilatkozatát: „Úgy gondolom, hogy az Atari név a cég legértékesebb vagyontárgya. Termékei, még a mostaniak is, elsőrendűek." Hja, az idők és a vállalati pozíciók változnak!

Tramiel nyilatkozatának megfelelően járt el. A meglévő termékekből felhalmozódott készleteket jelentős árengedménnyel árusította ki. Időközben kidolgozta a régi konstrukciók új, kiválóan formatervezett és nagyobb teljesítményű modelljeit. Az 1985. évi karácsonyi vásáron már ezzel akarta tartani piaci pozícióit. Egyúttal új piaci pozíciót kívánt szerezni Macintosh-szerű, 520ST gépével. A teljes kiépítésben mindössze 800 dollárért kínált gépet 1985 augusztusában dobta piacra. A Digital Research GEM nevezetű szoftver rendszerére építő gépét a háziszámítógépek felső kategóriájának, illetve az üzleti/professzionális gépek alsó kategóriáinak piacára szánták.

Az Atari az első két hónap után 50 ezer 520ST gép eladásáról beszélt, ezt azonban a kereskedelem adatai nem látszottak megerősíteni. Hogy Tramielnek elég nagy gondjai lehetnek, az abból is látszik, hogy szeptember végén két helyettese is kilépett a vállalkozásból. Nem igérkezik könnyűnek ez a feltámadás!

Jobs bosszúja

Tramielnek nyilván az volt az egyik szándéka, hogy régi vállalatának annyi borsot törjön az orra alá, amennyit csak lehet. Nem rövid távon tervezi bosszúját, hiszen tudja, hogy az igazi piaci sikerhez idő kell. Steve Jobs sem kevésbé boszszúálló alkat, ezt már ama ominózus kijelentése is mutatta, amit leváltása során tett: „A csata elveszett, de a háború még nem fejeződött be." Mentsége, hogy meglehetősen kíméletlenül bántak el vele.

A cégen belül teljes mértékben feleslegessé vált. Nem voltak napi feladatai, tulajdonképpen a kutya sem törődött vele. Minden arra irányuló kísérlete, hogy valami érdemi feladatkört kapjon, környezete teljes közönyével találta magát szembe. Steve Jobs a teljes lelki krízis állapotába jutott. Harmincévesen cél nélkülivé lett az élete. Hiába volt 100 millió dollár feletti részvényvagyona, rá kellett jönnie, hogy mit sem ér a vagyon, ha nem foglalkozhat azzal, amire képességei őt alkalmassá teszik.

Mikroinformatikai forradalom Lengyelországban

Három hónapra volt szüksége ahhoz, hogy ismét összeszedje magát, és ekkor bombaként robbant a hír: Steve Jobs kilépett az Apple-től, lemondott az igazgatótanács elnöki posztjáról, és öt másik kollégájával új vállalkozásba kezd. Az Apple rögtön lépett. Ötmillió dollárra perelte be Jobsot, amiért az Apple-nél vezető állást betöltő társai toborzásával jelentős kárt okozott a cégnek, új vállalkozásában pedig az Apple know-how felhasználásával egy konkurens gyártmányt akar piacra hozni.

Jobs váltig tagadja az ellene felhozott vádakat, és azt állítja, hogy a kis csapatnak még nincsenek megfelelően kidolgozott elképzelései. Egyedüli ötletük az, hogy az egyetemeknek kellene olyan szuper teljesítményű személyi számítógépeket gyártani, amelyekkel a legkülönbözőbb tudományos kísérletek szimulált bemutatását támogathatnák. A külső megfigyelők éles szemét ugyanakkor nem kerülte el az a tény, hogy Jobs távozási akciója éppen az új Apple első jelentős piaci bejelentésének időpontjára esett.

A szolid, megalapozott üzletmenetre áttérő cég egy sor gyártmányt jelentett be az Apple II vonal megerősítésére és a Macintosh perifériális kiépíthetőségének növelésére. Ekkor közölte az Apple azt is, hogy Macintosh gépét a közeljövőben nyitott konstrukciójúvá fogja tenni. Ezek mind olyan bejelentések voltak, amelyek a Jobs által korábban elkövetett piaci hibákat korrigálták. Az előkészített hírverést torpedózta meg utolsó akciójával Jobs, aki furcsa módon még mindig az Apple legnagyobb részvényese a maga 9 százalékával. Az új vállalkozáshoz ugyanis elég neki az a 27 millió dollár, amit 2% részvény eladásával szerzett.

●

Új időszak kezdődik ebben az évben vállalkozási szempontból is a mikro-számítástechnikában. Nem valószínű, hogy az előző időszakéhoz hasonló szerephez juthatnak a nagy vállakozói egyéniségek ebben az új szakaszban. A mai piacon csak a kollektív, maximális együttműködésre építő vállalati munkával lehet sikereket elérni. Ez a fő tanulsága ezeknek a rendkívüli történeteknek. Olyan tanulság, amit hazánkban is illene megszívlelni.

NACSA SÁNDOR

A hetvenes években Lengyelországban a zsebszámológépek óriási népszerűségre tettek szert. A nyugati országokba irányuló turistaforgalom jelentős mennyiségű áru beáramlását eredményezte, jóllehet — a magas feketepiaci valutaárfolyam miatt — nem olcsón. A behozott eszközök a hazai szabadpiacon csillagászati áron cseréltek gazdát. A vevőkör a műszaki értelmiségre és egyetemistákra szűkült. Az újabb modellek iránti kereslet mindig akkor nőtt, amikor áruk a mérnöki fizetés szintjére csökkent.

Az előző évtized végére a programozható zsebszámológépek kerültek az érdeklődés központjába. Sokaknak ez volt az első 'kézzelfogható' találkozásuk a megoldandó feladat algoritmizálási igényével. Addig a mérnökök túlnyomó többsége számára a számítástechnikai ismeretek szerzése egy vagy két féléves FORTRAN programozási tanfolyam után bevégződött, még az egyetemi évek alatt. A számítóközpontok nehézkessége alaposan leszűkítette a számítógépek segítségével megoldható feladatok körét.

A programozható zsebszámológépek elsőként ütöttek rést a számítástechnikával kapcsolatos idegenkedés és kiábrándultság falán, bár figyelembe kell venni azt is, hogy nagyobb részük inkább a birtoklási ambíciók kielégítését szolgálta, mintsem konkrét alkalmazásokat. A korábban tágas piac ezért rövid időn belül telítődött. Másrészt az 1980-ban elkezdődött társadalmi változások körüli általános vita is háttérbe szorította a műszaki újdonságok iránti érdeklődést. Ezt a tendenciát csak erősítette az 1981 végén bevezetett szükségállapot és a gazdasági válság rohamos elmélyülése.

A gazdasági embargó és a nyugati információk áramlásának korlátozása lényegesen késleltette az olcsó házi mikroszámítógépek (ZX80, ZX81) megjelenését. A külföldi kiutazási lehetőségek is leszűkültek, így az első mikroszámítógépek csak 1983-ban jelentek meg Lengyelországban. A csekély kínálat és a nagy kereslet miatt a szabadpiaci árak az egekig emelkedtek; a vevőkör néhány lelkes amatőrre és a magánszektorban dolgozó műszaki értelmiségre korlátozódott.

Ez a kezdetben marginális jelenség a ZX—Spectrum megjelenése után csak erősödött. Ez a gép ugyanis elődjeihez képest lényegesen jobb grafikai lehetőségekkel rendelkezik, ezért rengeteg látványos játékprogram készült rá, és éppen ezek a játékprogramok voltak azok, amelyek széles társadalmi érdeklődést keltettek. Ezt csak fokozta a televízió, amelynek ismeretterjesztő műsorai sok helyet szenteltek számítógépes — köztük Spectrumra írt — játékprogramoknak is.

Később a magán-valutaszámlák részleges befagyasztása megnehezítette a nyugati csomagküldő cégeknél történő megrendeléseket. Ennél is nagyobb korlátozást jelentett azonban a lengyel vámtörvény. Az engedélyek kiadása nehézkesen ment; a vámhatóságok nem tudták, hogy a számítógép veszélyt vagy társadalmi hasznot jelent-e. Csak tudós szakemberek szűk csoportjának a kérvényeit vették figyelembe, akik a számítógép-vásárlás szükségességét kemény érvekkel tudták alátámasztani. Sokan megpróbálták bebizonyítani a vámhatóság előtt, hogy a külföldről behozott mikroszámítógép csupán tv-játék.

Az emberek elszántságát legjobban jellemzi az a tény, hogy míg a magánkézben lévő mikroszámítógépek számát több ezerre becsülték, a vámhivatal alig több, mint száz behozatali engedélyt adott ki. A mikroszámítógépes divat növelte az elavult vámszabályok okozta társadalmi elégedetlenséget. A magas szabadpiaci árak okát szintén a vámban keresték. A sajtóban nyíltan kifogásolták a műszaki fejlődés akadályozását.

A széles körű kritikának sokáig semmilyen visszhangja nem volt. Csak amikor a mikroszámítógépekért folyó harcba nagy tudományos tekintélynek örvendő személyek is bekapcsolódtak, engedett végre a társadalmi nyomásnak a Központi Vámhivatal elnöke. A behozatali engedélyt ma már automatikusan adják ki az elvámolandó áru bemutatásakor. A tényleges vám mértéke 100 zloty/kg, vagyis inkább eszmei jellegű.

Így aztán 1984. december közepétől szinte elárasztották a mikroszámítógépek az országot. Majdnem minden külföldről visszatérő személy ZX—Spectrumot vagy C64-et hozott magával — igaz sokan csupán a nagy nyereség reményében. Mindez mégis a szabadpiaci árak rohamos csökkenéséhez vezetett. A ZX—Spectrum ára például 160-180 ezer zlotyról 100-120 ezer zlotyra mérséklődött. A jelenleg legnépszerűbb mikroszámítógép-típusok a ZX—Spectrum és a Commodore 64.

A ZX—Spectrumokkal együtt megjelentek a Nyugatról behozott programok is. Másolás és csere útján sok felhasználó több száz programból álló könyvtárhoz jutott. A programmásolás az átlagfelhasználónak nem okoz nagyobb nehézséget, a dokumentáció (használati utasítás) másolása azonban igen. Sokaknak van például assembler fordítóprogramjuk (néha több is, szövegszerkesztővel és disassembler programmal egybeépítve), de nem tudják használni leírás hiányában.

A nyugati piac új, hasonló árú, de gyakran jobb jellemzőkkel rendelkező mikroszámítógépei Lengyelországban mérsékelt érdeklődést váltottak ki. A számítógép felhasználója ugyanis jól tudja, hogy maga a hardver még nem old meg semmit. A ZX—Spectrumra készült nagy mennyiségű szoftver ezt a típust részesítette előnyben.

A lengyel nyelvű mikroinformatikai szakirodalom hiánya, érdekes, de tartalmilag félrevezető tévéműsorok (egymás mellett mutatták például a CAD — számítógéppel támogatott tervezés — alkalmazásait és a ZX—Spectrumra megírt, jó grafikájú játékprogramot, de azt elfelejtették megemlíteni, hogy egészen más osztályú eszközökről van itt szó) sokszor csalódáshoz vezettek. Gyakori eset, hogy valaki ZX—Spectrumot vásárol professzionális célokra, például adatbázisok létrehozásához, számítógéppel támogatott tervezéshez. Rövid örömteli időszak után, amikor a család és az ismerősök önfeledten ját-

szanak, a tulajdonos észhez tér: olyan programokat keres, amelyek segítségével megvalósulnának a munkahely forradalmi változásairól szóló mítoszok. Néhányan — elkésve ugyan — szakember tanácsát kérik. Akkor tudják meg, hogy a játékprogramok színes ábrái a hardverkorlátok figyelembevételével készültek, a gyakorlatban viszont nehéz lenne például csak olyan mechanikai alkatrészeket tervezni, amelyek még megjeleníthetők a képernyőn. Továbbá a kazettás magnón tárolt adatbázis ritkán jelent igazi hasznot, mert a kezeléséhez szükséges idő összeméréhető a hagyományos, papír-ceruza megoldás időigényével.

A szakemberrel való kapcsolat, bár kiábrándulást okoz, mégis hasznos. A felhasználó megérti, hogy tervei nem álmodozások, mert megvalósíthatók drágább, a közeljövőben talán elérhető, más osztályú eszközökkel. A megvásárolt mikroszámítógép mégsem értelmetlen beruházás; a meglévő felhasználói programok a módszert teszik érthetővé, széles körű alkalmazásokat ismertetnek.

Szakember- és ismerethiány következtében sok felhasználó saját szakállára tevékenykedik. Tapasztalatokat gyűjt, és amikor újból csalódás éri, az elfecsérelt idő és erőfeszítés mélyíti kiábrándultságát. Sokan egyre inkább azt hiszik, hogy a mikroszámítógépek igazán csak szórakoztató célokat szolgálhatnak, erre a célra viszont túl drágák.

A mikroszámítógép (még ha közvetlenül Nyugatról hozták is be) árának összehasonlítása az átlagfizetéssel arra a következtetésre juttat, hogy csak az elit számára elérhető berendezésről van szó. Elgondolkoztató tehát a Lengyelországba érkező mikroszámítógépek mennyisége, bár a ZX—Spectrum szabadpiaci ára még mindig alacsonyabb a hazai gyártmányú színes tévékészülék áránál. Az érdeklődők széles körének, főleg az ifjúságnak nincs lehetősége mikroszámítógép-vásárlásra. Természetes tehát a mikroszámítógépes klubok, szakkörök létrehozására irányuló törekvés.

Az első, Abakus nevet viselő mikroszámítógépes klub 1983 tavaszán kezdte működését, vagyis a mikroszámítógépes őrület kezdeti időszakában. A klub lelkes elnökének köszönheti létezését, aki megteremtette a működési feltételeket: helyiséget és néhány mikroszámítógépet szerzett. Az alapító szándékában állt megtalálni azokat a fiatalokat, akiknek dús a fantáziájuk, és képesek is azt olyan számítógépes játékok írására felhasználni, amelyek az oktatásban is alkalmazhatók. A klub nagy érdeklődésnek örvendett. Sokan voltak azonban, akiket csupán a billentyűütögetés érdekelt, ezért ezeknek a gépekhez való hozzáférését korlátozták, előnyben részesítve azokat, akik nemcsak az idegen galaxisból származó ellenségre vadásztak, hanem komolyabb feladatokat is vállaltak.

Az Abakus után újabb klubok alakultak. A legnagyobb problémájuk természetesen a berendezések megszerzése: a gépek drágák, nehezen hozzáférhetők, igen nehezen számlázhatók, főleg állami intézmények által támogatott klubok esetén, mivel azok magánszemélytől nem vásárolhatnak. Léteznek ugyan közvetítő cégek, de még a szabadpiaci árnál is többet kérnek az eladott berendezésért. A magánszemélytől történő vásárlást az adótörvények is nehezítik.

Ezek a problémák más, fontosabb kérdéseket szorítanak háttérbe. Mire végre sikerül a klubvezetőnek az annyira óhajtott mikroszámítógépet beszerezni, kiderül, hogy senkinek még elképzelése sincs a klubtevékenység programjáról, koncepcióiról. Végül a gépeket — jobb híján — mégiscsak játékra használják.

Valamivel jobban működnek azok a klubok, amelyek azonos típusú mikroszámítógépek tulajdonosait fogják össze. Talán túlzás is ezeket klubnak nevezni; egyszerűen néhány ember spontán társaságot alapít, ami megkönnyíti a programok, dokumentációk, könyvek stb. beszerzését. A konkrét szükségletek hasznos tevékenységet eredményeznek. E társaságok tagjai általában hasonló tapasztalatokkal és tudással rendelkeznek. Közös érdeklődés keretében ezt a tudást ugyan bővítik, de az ismeretek terjedése így is korlátozott.

Kivételnek számít itt az Apple gépcsalád felhasználói klubja. Tagjai az ország különböző területeiről származó magántulajdonosok és a munkahelyi gépet használó dolgozók. Közös jellemzőjük a hobbi jellegű felfogás, ami ebben az esetben bizonyos szenvedélyt jelent és nem az ismeretek hiányát. Az Apple felhasználói klub a NOT (Műszaki-Tudományos Egyesület) mikroprocesszor-felhasználók klubja támogatásával jött létre, elég különleges módon. A havonta megjelenő INFORMATYKA című folyóirat mikroinformatikai melléklete, a mikroKLAN háromrészes cikket közölt az Apple mikroszámítógépről, amely gép történelmi jelentőségű a mikro-számítástechnika fejlődésében. A cikk szerzője, aki maga is Apple II mikroszámítógép tulajdonosa, a cikkben a telefonszámát is megadta. Ez könnyű kapcsolatteremtésre adott lehetőséget több tucat Apple-felhasználó számára. Így jött létre a közös érdekeltségű csoport, vagyis a klub.

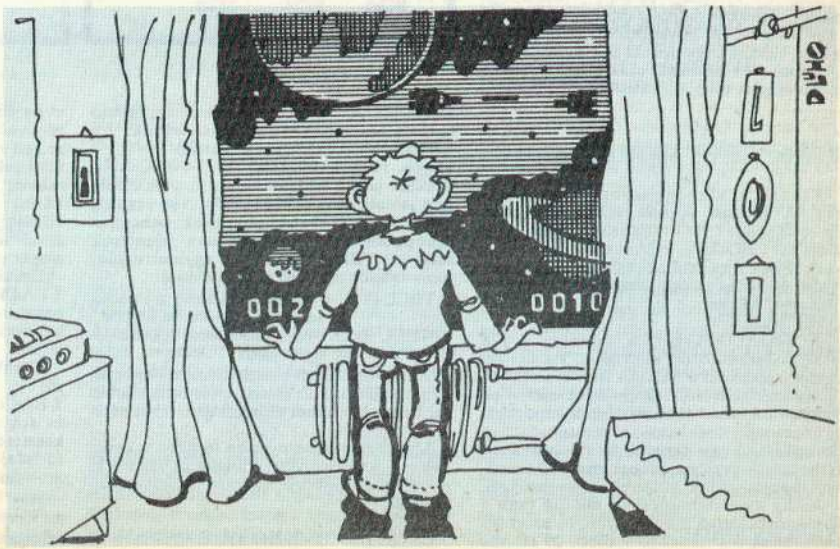
Néhány éve megfigyelhetjük, hogy az ifjúsági szervezetek olyan új működési formákat próbálnak kidolgozni, amelyek vonzó hatással lennének a fiatalokra. A miroinformatika lehetőségeit elsőként a Lengyel Cserkészszövetség (ZHP) ismerte fel. A Lengyel Rádió II. műsorában sugárzott cserkészmagazin keretében helyet kapott néhány mikroinformatikai műsor is. Érdekes kísérletet végeztek: a rádióadásban URH-n egy rövid játékprogramot közvetítettek olyan formában, ahogyan azt a ZX—Spectrum a magnókazettán tárolja. A hallgatók érdeklődése igazolta az ötlet helyességét. Az országban kezdtek létrejönni Informik nevet viselő cserkész mikroinformatikai klubok.

A cserkészszövetség után a Lengyel Szocialista Ifjúsági Szövetség is tervezi, hogy a meglévő ismeretterjesztő műszaki klubokra támaszkodva mikroinformatikai szakköröket hoz létre.

A számítástechnika népszerűsítése terén legkevésbé tevékeny a Lengyel Diákszövetség. Néhány alapszervezet rendelkezik ugyan mikroszámítógéppel; ez érdekes és kifizetődő munkalehetőséget ad egyetemistáknak. Példaképpen a szövetség utolsó közgyűlésén az információs szolgálatot mikroszámítógépek segítették.

Mindez azt mutatja, hogy Lengyelországban a mikroszámítógép már nem elérhetetlen álom, ködös mítosz. Bár használata főleg a nagyvárosokban terjed, a társadalmi tudatban bizonyos tartós fejlődést is okozott. E társadalmi tudati fejlődés megváltoztathatja a hatóságoknak az új technikához való óvatos hozzáállását. A számítástechnika, amely a házi Spectrumnak köszönhetően elveszítette varázslat jellegét, utat tör a professzionális alkalmazásoknak.

ANDRZEJ J. PIOTROWSKI



A LOGO új változata, a SmartLOGO

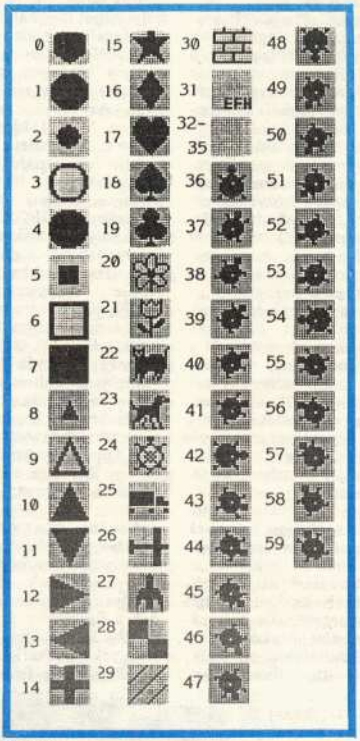
A számítástechnika oktatói, a számítástechnikát felhasználó pedagógusok véleménye megoszlik abban a kérdésben, hogy a reflex- és ügyességfejlesztő játékok minden határon túli használata milyen kihatással van a diákok számítástechnikához való affinitására. Egyesek szerint a gombok „ész nélküli" nyomogatása eltereli a diákok figyelmét arról, hogy valójában milyen eszközzel is állnak szemben, mások szerint éppen ez az egyik útja a számítógéphez való közeledésnek.

Egy dologban azonban minden, a számítástechnika oktatási célú vizsgálatával foglalkozó szakember egyetért: saját játékprogramok készítése mind a problémamegoldó készség, mind a számítástechnikai szakmai ismeretek fejlesztése szempontjából a legkiválóbb módszer, mivel ilyenkor a diák saját igényeihez keresi a legmegfelelőbb szakmai megoldást.

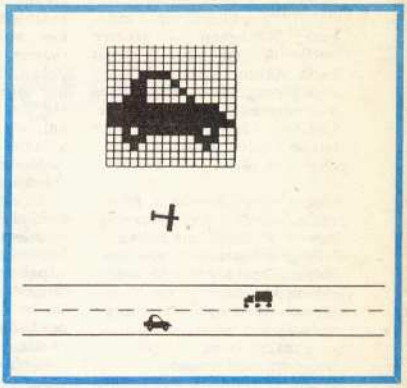
Az ilyen irányú törekvések segítésére készült a LOGO nyelv, ennek is egy továbbfejlesztett nyelvjárása, a SmartLOGO, ami gyors sikerélményt nyújt látványos képernyőrajzok megjelenítésében és mozgatásában.

A LOGO nyelv alapváltozatát már több cikkünk ismertette, és a nyelv oktatásra gyakorolt hatásával is foglalkoztunk előző számainkban. Most a SmartLOGO azon részeit kívánjuk ismertetni, amelyek a „régi" LOGO-hoz képest alapvetően új szolgáltatások megvalósítására készültek, és amelyek segítségével a programozó egy új LOGO-világba juthat el.

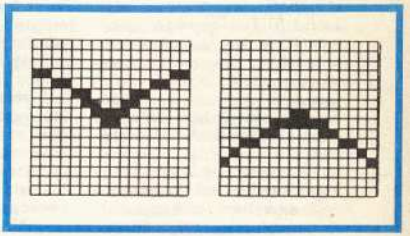
A SmartLOGO-t az MIT kutatócsoportja fejlesztette ki Adam és Apple számítógépekre. További implementációi folyamatban vannak. A következőkben az Adam gépre történt implementációt vesszük alapul, ami más gépeknél csak a hardver adta sajátosságok alapján jelenthet eltéréseket.



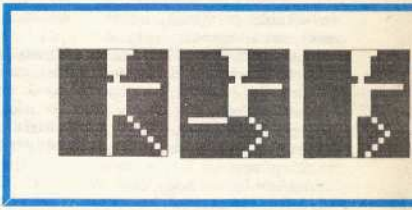
1. ábra



2. ábra



3. ábra



Színes, mozgó ábrák világa

Kezdjük el a mozgatást úgy, hogy a képernyőn látható teknőc sebességét 0-ról 25-re állítjuk:

```
? SETSPEED 25
```

A teknőc újra és újra végighalad a képernyőn, és addig mozog, ameddig sebességét 0-ra nem állítjuk.

```
? SETSPEED 0
```

A SmartLOGO-ban a sebesség −128 +128 értékek között mozoghat. Pozitív sebességértékeknél a haladás iránya a teknőc állása szerint előre, negatív érték esetén ellentétes irányban történik. Így a képernyőt akár mozgó méhkassá varázsolhatjuk a SmartLOGO egy alapvető tulajdonságának felhasználásával; egyidőben max. 30 teknőcöt mozgathatunk különböző sebességgel.

Eddig alapértelmezésben mindig csak a 0. teknőcöt láttuk a képernyőn, és ezzel végezhettünk műveleteket, a többi 29 teknőc rejtve maradt. Pedig könnyűszerrel tehető aktívvá bármelyik közülük. Ha leírjuk

```
? TELL 1 ST (ST=Show Turtle)
```

vagyis, hogy „MONDD AZ 1-NEK, MUTASSA MEG MAGÁT", az 1. számú teknőc jelenik meg a képernyőn. Vigyázzunk, előtte jó, ha a 0-st elmozgatjuk, nehogy a középpontban egymást takarva jelenjenek meg. A TELL paranccsal egyszerre bárhány teknőc aktivizálható, például:

```
? TELL [2 3 4] az aktivizálandó teknőcök
                sorszáma listába rendezett
? TELL ALL      minden létező teknőc legyen aktív
```

Ezek után minden aktív teknőc megkezdheti a saját önálló életét; elfordulhat jobbra, balra, lépdelhet előre, hátra, lehet saját sebessége stb.

Unalmas a képernyő, ha minden teknőcnek azonos színe van. Állítsuk az aktív teknőcöket különböző színűre a következő módon:

```
? TELL ALL
? EACH [FD 10*WHO SETC WHO]
```

Az EACH parancs jelzi, hogy a következő utasításlistát minden aktív teknőcre végre kell hajtani. A WHO operátor az EACH által pillanatnyilag kiválasztott teknőc sorszámát adja vissza (jelen esetben 0 és 29 között minden szám). Mivel összesen 16-féle szín létezik, két-két teknőc azonos színű lesz. A színkódok a 16-tal történő osztás maradékaként alakulnak ki.

Ezután, ha még a

```
? EACH [SETSP 10 WHO]
```

parancsot is kiadjuk, nem győzzük majd kapkodni a fejünket a nagy összevisszaság láttán.

Töröljük is le a sok teknőcöt a képernyőről

```
? EACH HT (HT=Hide Turtle)
```

és fogjunk egy értelmesebb képsor szerkesztésébe. Ehhez előbb a SmartLOGO egy új tulajdonságával kell megismerkednünk, a 60-féle képecske fogalmával.

A számítógép egy meghatározott tártartományában 60-féle előre megszerkesztett képecske van elhelyezve. Ezekre a sorszá-

muk szerint lehet hivatkozni, s formájuk a képszerkesztő segítségével bármikor megváltoztatható. Az *1. ábra* mutatja a 60 képecske kezdőértékét. A kezdőértékre azért van szükség, hogy a felhasználónak csak akkor kelljen ábraszerkesztéssel foglalkoznia, ha ez a menü nem elégíti ki igényeit. A 32—35. képek üresek, először ezekre érdemes új rajzot szerkeszteni, így nem romlik el egy már meglévő kép.

A megszerkesztett képecskék bármelyike egy-egy teknőchöz rendelhető hozzá. Alapértelmezésben, mikor egy teknőc aktívvá válik, a 36. kép rendelődik hozzá. Elforgatáskor a 36—59. képek közül a megfelelőre vált.

A meglévő készletből állítsunk elő egy mozgó képet. Alul az úton haladjon két teherautó egymással szemben, felül az égen egy repülőgép. A feladatot eljárások sorozatával építjük fel.

```
TO AIR
TELL 1 SETC 14
SETSH 26
SETY 75
SETSPEED 8Ø
END

TO TRUCKS
TELL [2 3]
EACH [SETSH 25]
ASK 2 [SETC 6 SETY -2Ø
SETSPEED 4Ø]
ASK 3 [SETC 1Ø SETY -6Ø LT 18Ø
SETSPEED 4Ø]
END

TO ROAD
TELL 4
HT
PU SETPOS [-123 -8Ø] PD FD 248
PU SETPOS [-123 Ø] PD FD 248
PU SETPOS [-123 -4Ø]
REPEAT 124/5 [PD FD 5 PU FD 5]
END

TO PICTURE
ROAD
TRACK
AIR
END
```

? PICTURE

Módosítsuk a képünket úgy, hogy az egyik teherautó helyett az úton személyautó közlekedjen. Mivel a személyautó ábrája a menüben nem szerepel, ezt a képet a képszerkesztővel létre kell hozni. A képszerkesztőt a 32. üres képre hívjuk meg:

? ES 32

Nevezzük el ezt a képet CAR-nak, hogy a későbbiekben ilyen néven hivatkozhassunk rá:

? NAME 32 "CAR"

Ezután a kialakított személyautó rajzát kicseréljük az egyik teherautóval, és a sebességét is növeljük.

```
TO TRAFFIC
PICTURE
TELL 3
SETSH:CAR
SETSPEED 6Ø
END
```

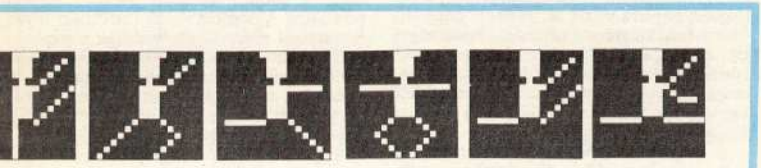
? CAR

Az eredmény „pillanatfelvétele" a *2. ábrán* látható.

Csináljunk mozit!

Az eddigi mozgó képek korlátja volt, hogy egy-egy részképecske önálló mozgásra (sebességre) volt ugyan képes, de alakját mozgása során nem változtatta. Most próbálkozzunk olyan mozgó képek előállításával, ahol alakváltozás is létrejön. A képszerkesztő segítségével rajzoljunk két madarat felfelé és lefelé tartott szárnyakkal *(3. ábra)*. Legyenek ezek a 10. és 11. képecskék.

4. ábra



Ezek után gépeljük be a következő egyszerű eljárást:

```
TO MADAR
SETSH 1Ø
WAIT 2Ø
SETSH 11
WAIT 2Ø
MADAR
END
```

Próbáljuk ki, mi történik:

? MADAR

A madarunk mozgatja a szárnyát! Ez már igazi animáció! Hogyan is értük ezt el? Egyszerűen a mozgó kép fázisrajzait válogattuk.

Nézzük meg ezt a módszert egy némileg bonyolultabb példán! Tervezzünk mozgó alakot, amely előbb tornagyakorlatokat hajt végre, majd végigszalad a képernyőn. Ehhez első lépésként a mozgás fázisrajzait kell megtervezni, majd a képszerkesztő segítségével ezeket például az első 9 képecskére elhelyezni *(4. ábra)*. Ezután a mozgatást a következő eljárással valósíthatjuk meg:

```
TO TORNA
SETX -1ØØ
MAKE "HELYBEN 5
REPEAT 3 [SETSH:HELYBEN+1
WAIT 2Ø]
REPEAT 4Ø [MAKE "FUT -
1 REPEAT 6 [SETSH:FUT+1
FD 1→ WAIT 1Ø]]
TORNA
END
```

? TORNA

Saját SmartLOGO C 64-en

A SmartLOGO-nak C64-es implementációja nem készült, de a gép hardverje és a LOGO C64-es Terrapin változata lehetővé teszi LOGO eljárások készítését a SmartLOGO bizonyos funkcióira. A C64 spritegrafikája használható fel elsődlegesen erre a célra. A sprite-ok felfoghatók egy-egy teknőcnek, amelyekből — mint ismeretes — egyszerre 8 lehet a képernyőn. A SmartLOGO-hoz hasonlóan itt is létezik a TELL LOGO primitív, amely egy adott sprite aktivizálására szolgál, és a WHO operátor, amely az aktív sprite sorszámát adja vissza. A sprite-ok adatterületén kialakíthatók a képecskék kezdőértékei, amelyek később a sprite-szerkesztővel változtathatók. A teknőcök (sprite-ok) sebességének változtatását szoftver úton lehet elérni. Ehhez először a WAIT eljárást kell elkészíteni:

```
TO WAIT :SECONDS
REPEAT :SECONDS*3ØØ[ ]
END
```

Ezután a mozgatás elérhető például a teknőc folytonos előreléptetésével, majd a sebességállítás a következő eljárással:

```
TO SETSPEED :KMPERH
IF :KMPERH=Ø STOP
IF :KMPERH<Ø [BK 1 WAIT
128-ABS :KMPERH] ELSE→
[FD 1 WAIT 128 - :KMPERH]
SETSPEED :KMPERH
END
```

A C64 hardver sajátosságai alapján egy időben csak egy sprite-ra adható ki a SETSPEED parancs (csak egy sprite lehet aktív), és a sprite így elért állandó sebessége csak BREAK-kel (CTRL G) állítható le. Ahhoz, hogy egyszerre több sprite-ot aktívvá tegyünk, assembler szintű beavatkozás szükséges. Az viszont könnyen megoldható, hogy az aktív sprite-okat folytonosan váltogassuk, s így az az illúzió kelthető, hogy egyszerre több képen hajtunk végre parancsokat. Természetesen ezzel a mozgás némileg lelassul és szakaszos lesz.

Érdemes még elkészíteni az EACH eljárást, amelynek segítségével az utána álló listában felsorolt sprite-okra parancssorozat adható ki, és az ASK eljárást, amellyel egy adott sprite-hoz parancssorozat rendelhető. Ezeknek az eljárásoknak a kialakítását már az olvasóra bízzuk.

SIEGLER VERA

A nyomtatástechnik

A nyomtatás valamilyen — általában festékes — nyomot hagyó formának a papírra (fémre, műanyagra stb.) nyomásával alakít ki a papíron (fémen, műanyagon stb.) valamilyen információ, főleg vizuális közvetítésére szolgáló betű-, ábra- stb. alakzatot. A kezdeti nyomtatók közönséges prések voltak. (Erre emlékeztet pl. a sajtó név is.) A technika fejlődésével a szükséges nyomás, ill. a nyomóerő nagysága egyre csökken ugyan, de egy minimális érték alá természetesen sohasem süllyedhet, hiszen a nyomóformának (alakjától függetlenül) érintkeznie kell azzal a felülettel, amelyen nyomot kell hagynia. A klasszikus nyomdai nyomtatók nyomógépek maradtak, rányomtak valamit valamire, így esetükben a név és a funkció összhangban van.

A számítástechnika a nyomdai nyomógépekkel sem áruk, sem nehézkességük miatt nem tudott mit kezdeni. A számológépek „nyomtatói", kezdetben a távgépíró és a villanyírógép, jelenként, később azonban a betűkerekes (betűhengeres) nyomtatók soronként hozták létre a nyomtatási képet. Ez utóbbit ezért sornyomtatónak nevezik. (A betűláncos nyomtatót is sornyomtatónak szokás hívni, de ennél egy sor nem a sort alkotó jelek egyszerre való létrehozásával alakul ki.)

Ezek a számítástechnikában használatos nyomtatók azonban már csak a nevükben nyomtatók, helyesebb volna őket nem nyomtató-, hanem leütőgépnek nevezni, ugyanis mindegyikben főszerepet játszik egy ütés jellegű mechanikai impulzus. (A sornyomtatóknál a papírt hátulról ütik hozzá a nyomásképet létrehozó mechanikai elemhez, ill. a festékszalaghoz, az ütésjelleg azonban ezek esetében is határozottan dominál.)

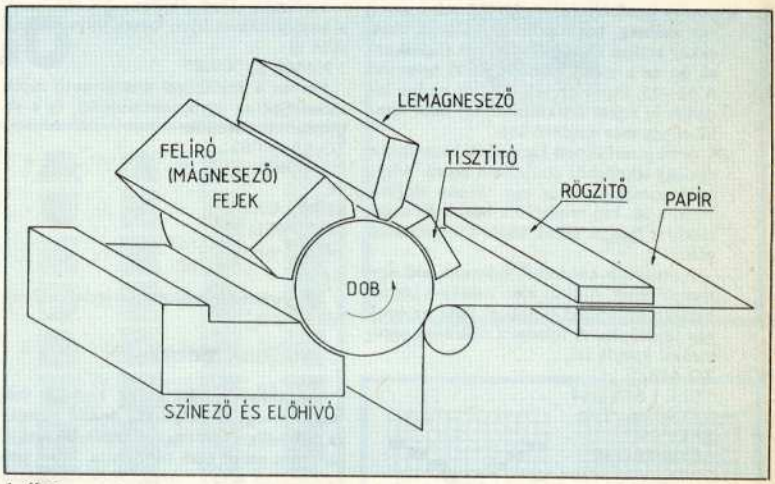
A nyomtatásképet létrehozó mechanikai elemeknek közvetlen vagy pl. festékszalagon keresztüli hozzáérése (hozzányomódása, hozzáütődése) a papírhoz egyes berendezésekben a fejlődés további folyamán is megmaradt úgy, hogy ezeket félig-meddig még jogos nyomtatóknak nevezni.

Vannak azonban olyan kimeneti perifériális egységek is a legfrissebb „nyomtató" választékban, amelyekben semmi szerepe sincs — vagy csak közvetett szerepe van — akár a klasszikus nyomásnak, akár pedig a leütésnek, a hozzáütésnek. Ennek ellenére — jobb szó hiján — nyomtatóknak hívjuk ezeket is, hiszen funkciójuk és szolgáltatásuk lényegében azonos vagy bővebb is, mint a nyomtatóké.

Négy ilyen „nyomtató" működési elvét ismertetjük a következőkben.

A mágneses nyomtató

A mágneses nyomtató elve a számítástechnika egyik régi tárolóberendezésének, a mágneses dobnak az elvéhez hasonlít. A körbe forgó dob mágnesezhető anyaggal bevont palástjára a magnetofonban játszódó folyamathoz teljesen hasonló (de itt digitális) módon, pontonkénti mágnesezéssel



1. ábra

írnak. Az írást egy több ezer írófejet tartalmazó fejmező végzi. A dobfelület mágnesezett pontjai magukhoz vonzzák a festékrészecskéket (amelyek anyaga mágnesezhető), a többi ponton a dobfelületre egyenletesen juttatott festékpor nem marad meg. Az így kialakított „nyomóhengerről" a festékport papírra viszik át. A papír pontjaira jutott festékrészecskéket hőhatással rögzítik, olvasztják rá a papírra. A forgódob folyamatos lemágnesezéséről és a festékpormaradványok eltávolításáról a megfelelő egységek természetesen állandóan gondoskodnak *(1. ábra)*.

A mágneses nyomtató írásképe nagyon szép. Ez nem csoda, ha figyelembe vesszük, hogy centiméterenként közel 100 pont létrehozására képes. A sebesség is figyelemre méltó, eléri a percenkénti 90 oldalt. A grafikus lehetőségek és az alkalmazható jelkészletek száma rendkívül nagy. Ennek a nyomtatónak nem okoz nehézséget az arab, de még a kínai írásjelek nyomtatása sem. Sok előnyös tulajdonsága miatt ez a típus várhatóan nem lesz tiszavirág-életű jelenség a számítástechnikában. Ára azonban ma még magas, személyi gépekhez csatlakoztatása elvileg problémamentes ugyan, de most még „drágább a leves, mint a hús". A felsőbb gépkategóriákban azonban egyes szakértők a mai „zajos nyomtatók" utódait látják bennük.

A lézernyomtató

A nyomtató egyik legfontosabb eleme egy forgó dob, amelynek palástját elektrosztatikusan folyamatosan feltöltik. Erre a feltöltött dobpalástra soronként ír a lézersugár. A lézersugár kívánt mozgatásáról egy eltérítő rendszer gondoskodik. Az „írás" lényege az, hogy a dobpalástnak a lézersugár érte pontjai kisülnek, ott megszűnik a (pozitív) feltöltöttségi állapot. A kisütött pontokon aztán meg tud tapadni a hengerre juttatott, előzőleg elektrosztatikusan feltöltött (pozitív töltésű!) festékpor, a többi ponton pedig az azonos töltések taszító kölcsönhatása miatt nem. Az így kialakított „nyomóhengerről" jut a porfesték aztán a dobhoz nyomott papír megfelelő pontjaira. A papírra került festékport rövid hőhatással rögzítik, ráolvasztják a papírra. A dob tisztítása és feltöltése ezalatt természetesen folyamatosan zajlik *(2. ábra)*.

Ezzel a technikával szép íráskép érhető el, emellett gazdag grafikus lehetőségek alkalmazására is módot ad.

A lézernyomtató már több éve a piacon van. Ára azonban még ma sem alacsony. Használata emiatt főleg nagyobb gépek mellett kifizetődő. Vannak azonban már olcsóbb személyi gépekhez kifejlesztett változatok is. Ezek sebessége legfeljebb 10 oldal percenként, szemben a nagyobb berendezések 200 oldal/percet is elérő sebességével. Ezek a gépek nem igényelnek különleges papírt, sőt egyéb felületekre, mint pl. műanyagra is nyomtathatunk velük.

A lézernyomtató elve nem új, hasonlít a száraz eljárásos fényképmásolás alapgondolatához.

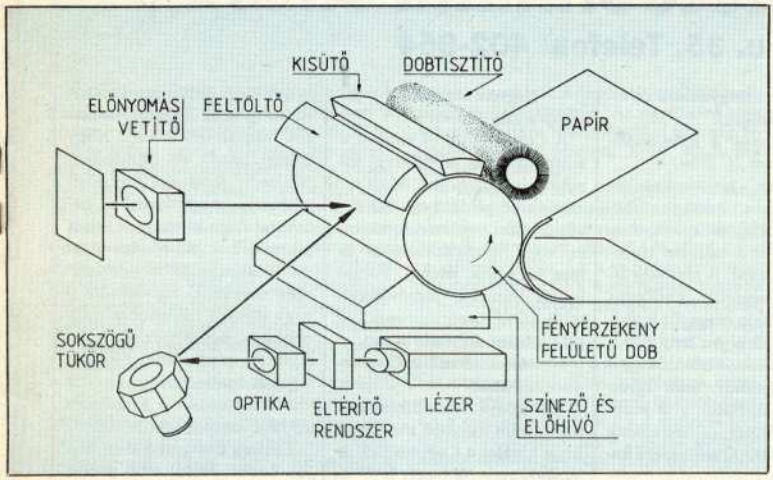
Ma úgy tűnik, hogy előnyei miatt e típus sem lesz rövid életű, hiszen nemcsak számítástechnikai területen használják. A nyomdatechnikában is működnek már modern nyomdai szedőgépekhez illesztett lézernyomtatók.

A kristályos folyadék nyomtatók

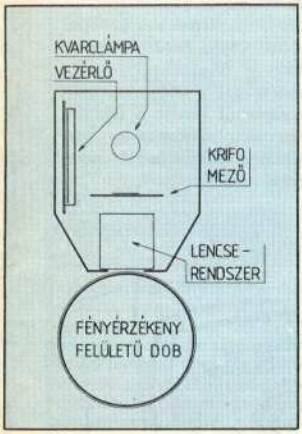
A kristályos folyadék (röviden krifo) kijelzők a karórákban és különösen a zsebszámológépekben, újabban pedig a személyi számológépekben tömeges elterjedtségnek örvendenek.

E nyomtatótípus a krifo kijelzőn megje-

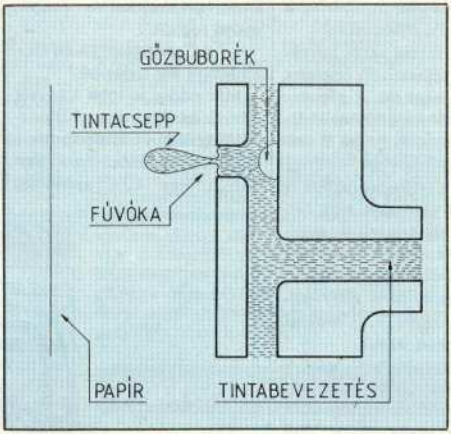
a újabb eredményei



2. ábra



3. ábra



4. ábra

lenő képet másolja át papírra, tisztán fotoelektromos közvetítéssel.

Mind az órában, mind a zsebgépeken tapasztalható a képváltás lomhasága. Ez a lomhaság a nyomtató sebességének versenyképességét veszélyeztető mértékű határt szabna. Ezért a nyomtatókban alkalmazott krifo megjelenítő képváltási sebessége az eddig megszokottnak kb. 200-szorosa. A kvarclámpával átvilágított rácsszerűen rendezett krifo képelemmezőből a fény egy lencserendszeren keresztül jut egy fényérzékeny hengerre. (A megjelenítő mezőről egy egész sor egyszerre kerül át e hengerre.) A fényérzékeny dob felületén levő, a környezőktől különböző tulajdonságúvá vált képpontok időlegesen színezőanyagot tapasztanak magukhoz, és ezt aztán — mint egy „nyomtatóhenger" — átadják a velük érintkezésbe hozott papírra. A papírra juttatott anyag rögzítése elvileg többféle módon is történhet. Az elv a közismert Xerox-berendezések elvével azonos *(3. ábra)*.

Az íráskép kiváló. A „felbontás" közel 100 pont/cm. Nyomtatási sebessége percenként 10 db A4-es formátumú oldal. Ez a nyomtatótípus már a személyi számológép tulajdonosok számára is elfogadható árú. Emiatt is hosszabb jövőt jósolhatunk e nyomtatótípusnak.

A tintacseppes nyomtató

Ez a nyomtatófajta az, amelynek működésében semmi szerepe nincs sem nyomásnak, sem ütésnek (ráütésnek). Jobb szó híján azonban ezt is nyomtatónak nevezzük, már csak azért is, mert ha produktumát (amely nem nyomtatvány!) nézzük, e típusnak csöppet sem kell szégyenkeznie az igazi nyomtatók előtt.

A *tintacseppes* nyomtató tintacseppeket, mikroszkopikus „pacákat" lő a papírra. Innen a neve is. A *tintasugaras* nyomtató név használata joggal kifogásolható, hiszen ezekben a készülékekben tintasugár sohasem alakul ki. Ez a nyomtató nem úgy ír, mint ahogy vízsugárral pl. egy vizes nyolcast írunk a járdára. Munkája tintacsepegtetés, csak nem függőlegesen, fentről lefelé, és nem a gravitációs erőre támaszkodva.

A tűs nyomtatókhoz hasonlóan a jelek itt is apró pontocskákból állnak össze. E pontocskákat azonban nem nyomótűk állítják elő a festékszalag papírhoz nyomásával, hanem a fúvócsövek tintacseppecskék papírra lövésével.

A tintacseppek kilövésére kétféle módszert alkalmaznak. Az egyik módszer a piezoelektromosságot használja. Rövid elektromos impulzussal állítják elő a tintacsepp kiröpítéséhez szükséges nyomásimpulzust.

Jó írásminőséget lehet előállítani az ilyen, milliméterenként kb. 10 pont (paca) sűrűséggel dolgozó berendezésekkel. Az írási sebesség 200-tól 600 jel/másodpercig terjed. Gyártanak 30-nál több fúvókás nyomtatókat is.

A másik módszer a tintacseppeket gőzbuborék létrehozásával löki ki a fúvócsőből. A rövid elektromos impulzus által keltett hő gőzt fejleszt, a gőzbuborék megnöveli a nyomást, ez a megnövekedett nyomás hajtja ki a fúvócsőben levő tintát, ami cseppé formálódva jut a papírra *(4. ábra)*.

A nyomtatási sebesség legalább 150 jel/sec, de megközelítheti ennek háromszorosát is. A centiméterenkénti pontszám megközelíti a százat. Így egészen jó írásminőség érhető el.

A rendkívül alacsony energiaigény miatt e típus táskaszámológépekben is használható.

Az összes bemutatott nyomtató szokatlanul csendes működésű. A tintacseppes nyomtató azonban mindegyiken túltesz, majdnem teljesen zajtalan. Hatalmas fejlődés eredménye ez. A kezdetben még a dobozba zártság ellenére is fülsiketítően hangos nyomtatók múzeumi tárgyakká váltak, és a zajosságuk miatt külön helyiségbe telepített sornyomtatók órái is meg vannak számlálva.

A tintacseppes nyomtatókhoz eleinte különleges papírra volt szükség. Ezek a nagy nedvszívó képességű papírok biztosították, hogy az egymás mellé került tintacseppek összefolyva folytonos vonalat is kialakíthassanak. A mai különleges tintákhoz azonban már nincs szükség speciális papírra. Sőt a papír mellett fémre, műanyagra is írhatunk velük. Több különböző fúvóka alkalmazásával pedig, amelyek más-más színű tintával dolgoznak, színes nyomtatást is tudunk végezni. Van olyan berendezés, amely 8 színnel több mint száz színárnyalat létrehozására is képes.

SZEBENSZKI SÁNDOR

2C Számítástechnikai Áruház

Budapest, 1136 Balzac u. 35. Telefon: 402-954

1985. szeptember 16. óta működik. Jelenlegi fő profilja Commodore 64 és 610 szoftverek értékesítése. A választékban több mint 130-féle terméket kínálnak. Ennek többsége feladatorientált felhasználói program, 5000 és 130 000 forint közötti árban, a tipikus ár 20 ezer és 30 ezer forint között változik. Széles a kínálat az általános célú felhasználói programokból, rendszer- és segédprogramokból is. Ezeket 3100 és 37 500 forint közötti áron lehet megvásárolni. A tipikus ár 10 ezer és 20 ezer forint között van. A Commodore 610-re kapható programrendszerek ára eléri a 200 ezer forintot, a tipikus ár 60 ezer és 90 ezer forint közötti.

Kaphatók különböző kiegészítők és könyvek is. A Novotrade saját fejlesztésű játékait mind kazettán, mind lemezen meg lehet vásárolni. A vevők szép kiállítású és kellően tájékoztató katalógusból választhatnak. A katalógust kívánságra postán megküldi a cég az érdeklődőknek. Az áruház alagsorában külön helyiség várja a fiatalokat, akik az ott elhelyezett Commodore gépeken szabadon játszhatnak vagy programozhatnak.

A 2C áruház megnyitása kapcsán néhány kérdést intéztünk Rényi Gábor igazgatóhoz.

— Mivel kívánnak elsősorban foglalkozni?

Olyan szoftverüzleteket szeretnénk fenntartani, amelyek a szoftvert, mint könyvet árusítják. A gyártók a fejlesztéssel és az áru előállításával foglalkozzanak. Nem lehet elvárni, hogy a felhasználó Budapestre járjon szoftvereket vásárolni. Még ötven ilyen üzletre lenne szükség az országban. Ehhez olyan partnereket keresünk, akik a 2C módszereivel, franchising konstrukcióban forgalmaznának. Erről már meg is kezdődtek a tárgyalások az első jelentkezőkkel.

— Mit nyújt a felhasználónak egy ilyen 2C üzlet?

Ez olyan kiskereskedelmi egység, ahol a termékek szabadon megtekinthetők, bemutatják, és a vevő kedvére kipróbálhatja azokat. A részletes áttekintést megfelelő katalógus segíti. Nemrég megnyílt áruházunk az első olyan hely az országban, ahol a vásárló megtalálhatja mindazt, amit keres. Nagyobb kínálatot — a jelenlegi kétszeresét — szeretnénk tartani. Szinte Quelle-szerű katalógust szándékozunk összeállítani.

— Milyen előnyt jelent ez a konstrukció a fejlesztőknek?

A gépek száma akkor növekszik, ha a felhasználó értelmes alkalmazásokat lát. A gép önmagáért való vásárlásának ideje lejárt. Olcsóbb árakra és szélesebb terítésre van szükség, hogy olyan felhasználókhoz is eljusson a gép, akik nem tartoznak az elkötelezett megszállottak közé. Így terjedhetnek igazán széles körben a fejlesztők termékei, és a nagy példányszám révén a gyártók is sokkal jobban járnak.

— Hogyan kívánják befolyásolni a fejlesztéseket?

A programok széles körű terjesztése révén széles körű visszacsatolást kapunk. Ha minőségi és szolgáltatási problémákat tapasztalunk, azokat eljuttatjuk a gyártónak, szélső esetben akár beszüntetjük a termék forgalmazását.

— A leasing tevékenység mennyiben tartozik hozzá az üzleti koncepcióhoz?

A beruházási pénzeszközöket kímélő megoldásokat is szeretnék érvényre juttatni. Ezért gépek leasingszerű értékesítésével is foglalkozni fogunk a 2C keretében. A 2C szomszédságában bizományi boltot nyitunk a régi gépek értékesítésére.

— Hogyan támogatják a jövőben a régi gépeket?

A Commodore 64 vonalat a szoftverkínálat bővítésével továbbra is támogatjuk. Most vezettünk be például egy 3D nevű grafikai szoftvercsomagot, és egy rajzdigitalizálóval is meg kívánunk jelenni. Forgalomba hozunk tárbővítőket, szoft-ROM cardridge-eket, és még lehetne folytatni a sort.

— Mik a terveik a Commodore 128-assal?

Két könyv van előkészületben. Amint nagyobb tömegben fogja kínálni a Commodore és hazánkban is több ezer lesz belőle, általunk fejlesztett és más hazai szoftvereket is forgalmazni fogunk.

— A Commodore 16-tal kapcsolatban mit terveznek?

Már eddig is több könyvet adtunk ki: Hetedhét, Felhasználói kézikönyv és Bevezetés a BASIC-programozásba. Tervezzük egy C16 assembler könyv kiadását. Segíteni kívánjuk a C16-os programcsomagok iskolai terjesztését. Nagy üzletet látunk az iskolákban. A kétezer gépet nyilvánvalóan többezres magánforgalmi behozatal fogja követni; ez óriási piacot jelent. Az iskolai oktatás hatékonysága is ugrásszerűen növekedni fog. Mindez a nagy tömegű importot is megnyithatja, ugyanis az ilyen kategóriájú gépek hazai gyártása teljesen értelmetlen.

— Kívánnak-e foglalkozni az IBM-kompatibilis PC-kkel?

Leasing konstrukcióban fogjuk árulni. Minél több gyártó és kiadóvállalat szoftveráruját szeretnénk forgalmazni; erről már megszülettek a megállapodások. Olyan hardverkiegészítőket is forgalmazni fogunk, amelyekkel a műszaki alkalmazásoktól kezdve a többfelhasználós rendszerekig, igen sokféle felhasználás előtt megnyílik az út.

A COMMODORE 16

A Commodore cég egy kifejezetten olcsó háziszámítógéppel jelent meg 1984-ben a piacon. 300 nyugatnémet márka alatti jelenlegi árával egyike a legolcsóbb gépeknek a Commodore 16. Ehhez képest képességei nem is olyan szűkre szabottak. Hogy mit tud? — az alábbiakból ez is kiderül.

Külső megjelenése a már jól megszokott Commodore-forma. Külön tápegysége nincs, az adaptert a hálózati csatlakozó foglalja magába, melynek hosszabb használat esetén sem jelentős a felmelegedése. Installálását a legamatőrebbek is hibátlanul el tudják végezni, mivel minden csatlakozója más.

Klaviatúrája némileg eltér a C64-esétől. Újdonság az ESC (escape) billentyű, mely különböző szerkesztési, képernyőkezelési, kurzormozgatási és scroll-funkciók megvalósítását egyszerűsíti, illetve újdonságnak számítanak a FLASH ON, FLASH OFF billentyűk, melyek a képernyő adott részének villogtatását váltják ki. Elhagyták a RESTORE billentyűt, a különleges karakterek és néhány funkcióbillentyű helye megváltozott. A 8 darab programozható funkcióbillentyű alapértelmezésben is él — a leggyakrabban használt BASIC utasításokat tartalmazza.

A kurzormozgató billentyűk eltérnek a C64-étől, mivel a különböző irányú mozgatásra 9 db, megfelelő feliratozással ellátott billentyűt építettek be.

A hardver

Mikroprocesszora a 7501-es (6502 kompatibilis), a címezhető memória 64 kbájt lehet, ebből 16 k RAM van alapkiépítésben, de bejelentkezéskor a gép csak 12 277 bájt szabad területet jelez. A ROM terület tartalmazza a BASIC interpretert, a karaktertáblázatot, az „üzemrendszert", a Kernal-ugrótáblázatot, a B/K címeket és a TED-CHIP regisztert, valamint az NMI, RES és IRQ-vektorokat.

A gép háromféleképpen tudja kezelni a képernyőt. Szöveges üzemmódban 25 sorban, soronként 40 karakter írható ki a képernyőn. A nagy felbontású grafikus üzemmódban 320 pont van soronként a képernyőn és 200, pontokból álló sor. Ezekben az üzemmódokban a képernyőn csak két szín (előtér és háttér) tudunk használni. További két szín használatát teszi lehetővé a több színű grafikus üzemmód. Ekkor azonban a pontsorokban igénybe vehető pontok száma a felére csökken, vagyis a 320 × 200-as felbontás helyett mindössze 160 × 200-as felbontás áll rendelkezésre. A grafikus üzemmódok speciális változatai a szövegmegjelenítéssel kombinált üzemmódok. Ekkor az alsó 40 pontsor helyett 5 szövegsort jelenít meg a gép a képernyőn. A színek 16 színből álló palettáról választhatók. Ezenkívül a fényerejük is állítható, összesen 8 fokozatban. Az egyébként igen sokrétű grafikus lehetőségekkel bíró gép jelentős korlátja viszont, hogy nincsenek hardver sprite-ok.

A képernyőkezelés üzemmódjának megfelelően változik a rendelkezésre álló RAM kiosztása. Az üzletekben kapható alapkiépítésű változat 16 kbájt RAM-jából 2 k a rendszermemória, 1 k a szöveges üzemmód színmemóriája és 1 k a szöveges üzemmód képernyő-memóriája számára fenntartott terület. Az ezt követő 12 k-t használhatja szöveges üzemmód esetén a BASIC. A grafikus üzemmódoknál a gép 10 k-t elvesz ebből a képernyő grafikus kezelésének céljaira. 1 k szükséges a fényerő beállításához, 1 k a színtáblázathoz és 8 k a grafikus képernyő-memóriához. Így a *grafikus üzemmódoknál mindössze 2 kbájt RAM áll a BASIC rendelkezésére.* Ez komolyabb használathoz rendkívül kevés! Ilyen esetekben ezért feltétlenül szükséges egy RAM bővítő beépítése. Egy 64 k-s bővítővel közel 60 kbájtra növekszik a BASIC rendelkezésére álló RAM-terület szöveges üzemmód esetén. Grafikus üzemmódnál ez megfelelően 10 k-val kisebb. A RAM bővítést persze külön kell megvásárolni, és így az eredetileg olcsónak tűnő gép már nem is olyan olcsó.

Három hanggenerátora (ebből 1 zajgenerátor) 4 oktáv terjedelemben (110—1575 Hz) programozható; lehetőség van keverésükre, valamint 7 fokozatú hangerőszabályzásukra, illetve időtartamuk meghatározására. A hangeffektusok, dallamok megalkotását nagyban segíti a géphez mellékelt kézikönyv függelékében található táblázat. Összességében azonban gyengébb a hanggenerátor, mint a C64-é.

A perifériák

A Commodore 16 esetében a gyártó csak részben valósította meg a C64 perifériákkal való kompatibilitás szerintünk fontos követelményét. A gép nem rendelkezik analóg bemenettel, egyéb tekintetben azonban hasonló lehetőségei vannak, mint a C64-nek.

A megjelenítő. Kétnormás színes tévé (a gép PAL-rendszerű jelet ad), és C1701 (1702, 1802 vagy 1803) típusú videomonitor alkalmazható, de természetesen fekete-fehér tévéhez is csatlakoztatható. Ez utóbbi esetben azonban a színes lehetőségeket csak a szürke különböző árnyalataival lehet kihasználni. Az RF-jelet az UHF 36-os csatornán adja. A monitort a videokimeneten keresztül kell csatlakoztatni; ehhez a csatlakozózsinórt külön kell beszerezni, a géphez nem mellékelik.

A magnetofon (VC—1531). Csak ez a típusú gyári készülék csatlakoztatható, közvetlen bemeneten keresztül.

Hajlékony mágneslemezes egység (VC—1541 vagy VC—1551). Ez már teljesen azonos a C64-hez illeszthető floppyval. A csatlakoztatás módja is megegyezik: soros buszon keresztül illeszthető a floppy és a nyomtató. Ezek maximális számáról azonban a kézikönyv nem tesz említést, de az alkalmazható BASIC-parancsokból azt a következtetést lehet levonni, hogy a C64-hez hasonlóan, floppyból több is illeszthető.

A sornyomtató. A Commodore soros interfészét támogató nyomtatókat lehet használni. Ilyenek a C64-en is használt Commodore nyomtatók (MPS—801, MPS—802, MPS—803 és a DPS 1101), illetve a független perifériagyártók által ezekhez a gépekhez kidolgozott berendezések (pl. Epson GX—80 a megfelelő interfésszel).

Egyéb perifériák. Közvetlenül illeszthető 2 db botkormány. A csatlakozók azonban teljesen gyártóspecifikusak, így vagy a cég botkormányait kell megvásárolni, vagy pedig külön csatlakozó átalakítót kell beszerezni. (Így lesz egy elvben olcsó gép mégis drága. A szerk.)

A szoftver

A Commodore 16-os alapszoftveréről nyugodtan kijelenthetjük, hogy igen korszerűnek számít az operációs rendszer nélkül működő személyi számítógépek között. Ez a fejlettség a TEDMON monitornak és a BASIC 3.5-ös verziójának köszönhető.

A TEDMON monitor

Segítségével közvetlenül assembler memonikus kódban programozható a gép, nincs szükség az egyes utasítások gépi kódjainak ismeretére. (Az igazi assembly programozáshoz elengedhetetlen címkék használata azonban nem lehetséges. A szerk.). Emellett megvan a lehetőség a *közvetlen gépi* kódú bevitelre is. A gépi kódú bevitel ellenőrzésére is felhasználható a monitor *disassembler* szolgáltatása. A monitor e három legfontosabb szolgáltatása mellett még a következőket nyújtja: két memóriaterület (munkaterület) összehasonlítása — a nem egyező címek kiírása; adott tárterület feltöltése megadott bájtszóval; gépi kódú program indítása megadott címről; adott memóriarész feltöltése ASCII vagy hexadecimális kódokkal; gépi kódú és assembler programok betöltése kazettáról vagy floppyról, illetve ezek kimentése, valamint ellenőrzése; a tár tartalmának hexadecimális vagy ASCII kódban való kiíratása; a processzorregiszterek (PC, SR, AC, XR, YR, SP) tartalmának megjelenítése (ezek egyébként a TEDMON-monitor meghívásakor automatikusan kiírásra kerülnek); és a tártartalom átmásolása egy másik tártartományba — így egy utasítással elmenthető és később visszahozható például a képernyőtartalom.

A BASIC 3.5 verziója

Felülről kompatíbilis a C64 BASIC 2.0 verziójával. Bővítésként — több más utasítás mellett — tulajdonképpen a CP/M lehetőségeit foglalja magába. Újdonsága a C64-hez képest, hogy a színkezelést, a hanggenerátor programozását és maximum 255 karakter terjedelmű képernyőrészletek mozgatását a nehézkes POKE-utasítások helyett egyszerűen paraméterezhető BASIC-utasítások végzik. A színkezelést végző COLOR utasításhoz például meg kell adni, hogy minek a színét kívánjuk változtatni. Az alap, a keret, a kurzor színe és 2 darab ún. többletszín határozható meg egymástól függetlenül, 16 különböző szín 8-8 árnyalatával, melyek különböző színhatásúak, így összesen 121 szín állítható elő egyidőben. A fekete csak egy árnyalattal szerepel ebben a számban.

A hanggenerátor programozására a SOUND utasítás szolgál, miután a VOL utasítással beállítottuk a kívánt hangerősséget. 7 fokozat közül választhatunk, a VOL 0 a hanggenerátor kikapcsolására szolgál. A hang a tévé hangszórójából hallható. A SOUND paraméterei: a csatorna száma (1—3), a hangmagasság kódja (a függelékben levő táblázatból megállapítható), valamint a megszólaltatott hang hangzásának időtartama. Maximum 21,5 percig lehet 1 hangot 1 utasítással megszólaltatni.

A 3.5 verzió fontos újdonsága a SSHAPE/GSHAPE utasításpár, amelyekkel a Commodore 64 sprite-kezelését tudjuk jóval egyszerűbben megvalósítani. Az SSHAPE utasítással egy karakterlánc (string) típusú változóba tehetjük a képernyő tetszőleges részletét, melyet a két átlósan szemközt levő pontja koordinátáival kell meghatározni. E képernyőrészlet maximális méretét a karakterláncjelleg korlátozza, azaz 255 karakternyi fér el benne. A fentiekből következik az is, hogy itt nincs megkötés a sprite alakjára — tetszőleges formájú téglalap vagy sor lehet, szemben a C64 24×21-es méretével. Nincs megkötés ezek számára vonatkozóan sem.

A karakterlánc-változóban levő képernyőrészletet a GSHAPE utasítással mozgathatjuk. Ehhez az utasításhoz a változónév mellett paraméterként csak az új hely bal felső sarkának koordinátáit kell megadni, illetve a megjelenítés (normál vagy inverz) és a már ott levő ábrákkal való fedés módját — mellyel „VAGY", „ÉS" és „KIZÁRÓ VAGY" kapcsolatba hozható, — kell meghatározni.

A Commodore 16 BASIC-verziója több újdonság mellett kibővített formában tartalmazza a Sinclair Spectrum gépnél már megismert rajzoló utasításokat is. A DRAW utasítás hatására egy pont rajzolható, illetve két pont közötti egyenes húzható a két szélső pont koordinátájának megadása révén. A DRAW utasítások egy utasításban láncolhatók, így tetszőleges geometriai alakzat rajzolható ki a töréspontok koordinátáinak megadásával. Az így rajzolt pontok és vonalak törlésére ugyanez az utasítás szolgál, paraméterváltoztatással.

A CIRCLE utasítás alkalmas kör, ellipszis, körszelet, illetve ezekbe rajzolható sokszögek megrajzolására, ezek megadott szögű elfordítására, melyhez a kör középpontjának koordinátáit, a két tengelyen a sugarakat; körszelet esetén a kezdeti és a végszöget; sokszög esetén a belső szög kiegészítő szögét, illetve az elfordítás szögét kell megadni.

Újdonság a BOX utasítás, mely átlósan szemközti csúcsai koordinátáival megadott téglalapot rajzol ki, azt tetszőlegesen adott szögben elfordítja, és a megfelelő paraméter megadásával a kívánt színűre színezi be. A PAINT utasítás segítségével valamely tetszőleges zárt geometriai alakzat színezhető ki a kívánt színre. Ha az alakzat nyitott, akkor az egész képernyő beszíneződik. Ehhez a kiszínezéshez elegendő az alakzat egyetlen tetszőleges belső pontjának koordinátáit megadni.

A C16 BASIC-je rendelkezik kurzorállítási lehetőségekkel: a grafikus képpont, ún. pixel kurzort a LOCATE, a karakter kurzort a CHAR utasítások állítják a megfelelő pozícióba. Az utóbbi egyúttal az adott helytől kezdődő szöveg kiírását is végzi (grafikus üzemmódban is), ahol utasításon belül határozható meg a normál vagy inverz módú kiírás.

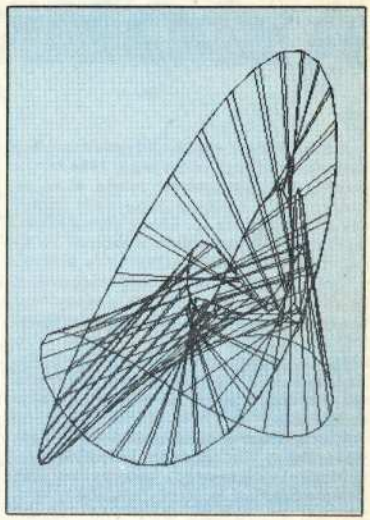
A RESTORE utasítás megadott sorszámú DATA sor elejére állítható.

A programozás könnyítését szolgálja a DO/LOOP/WHILE/UNTIL/EXIT utasításegyüttes. Újdonság a C64 utasításkészletéhez képest a GETKEY, mely annyiban tér el más gépek INKEY$ utasításától, hogy csak ténylegesen lenyomott billentyűt vesz figyelembe; ha semmit nem nyomnak le, akkor vár.

Kiváló PRINT USING utasítás áll rendelkezésünkre, melynél a négy leggyakrabban használt jel menet közbeni átírására a PUDEF utasítás szolgál.

A BASIC-parancsok közül említést érdemel az automatikus sorszámozást lehetővé tevő AUTO parancs és az átszámozást végrehajtó RENUMBER. A gép egyébként képernyő- (full-screen) editorral ellátott, amelynél lehetőség van az utasítássorszámok átírására is — vigyázni kell azonban arra, hogy a régi sorszámot DELETE-tel törölni kell, mert egyébként benne marad a programban. A RENAME parancs szolgál a lemezfájl nevének átírására; a BACKUP paranccsal egy lemez összes fájlja, a COPY-val egy fájlja átmásolható egy másik lemezre; a COLLECT befejezetlen fájlok lezárását végzi. Külön parancsok szolgálnak a BASIC-programok betöltésére és kimentésére lemez(ről)re, illetve kazettá(ról)ra (DLOAD, LOAD, DSAVE, SAVE), ezek ellenőrzésére (VERIFY). A lemeztartalom vagy megadott fájl listázására a DIRECTORY parancs használható.

A nyolc darab funkcióbillentyű tartalmának átírására szolgál a KEY parancs, amelynek segítségével egy utasítássor maximálisan lehetséges hosszáig (80 karakter) tetszőleges számú utasítást rendelhetünk a kiválasztott funkcióbillentyűhöz, amely a gép kikapcsolásáig vagy újabb átírásig érvényben marad, és a billentyű lenyomására végrehajtásra kerül. Bekapcsoláskor minden esetben a gyárilag generált parancsok, utasítások élnek.



A gép a Commodore 64-hez hasonló beépített órával rendelkezik.

A TRAP utasítás hatására nyomkövetés történik, mely abban tér el az egyébként itt is használható TRON utasítástól, hogy hiba esetén megadott utasításszorra ugorhatunk, így megjeleníthető a hibás sor sorszáma és a hibajelzés; más gépek ON ERROR funkcióját látja el.

A paraméterezhető vagy címtől indítható gépi kódú szubrutinhívás utasításszintről is végezhető.

Numerikus függvényei a C64-nél is megtalálhatók mellett tartalmazzák a hexadecimálisról decimálisra és vissza átszámoló DEC és HEX függvényeket; a botkormányokat értelmező JOY; a keret, a háttér, a kurzor és a többletszínek aktuális színkódját visszaadó RCLR, ugyanezek színintenzitásának kódját tartalmazó RLUM; a pixel kurzor aktuális helyzetét visszaadó DROT; a 4-féle grafikus mód közül az aktuális kódját visszaadó RGR függvényeket. Ki kell emelni az RND függvényt, melynek véletlenszerű kezdéséhez nem szükséges RANDOMIZE utasítás — enélkül is véletlenszerűen indítja a véletlenszám-generálást.

Karakterláncfüggvényei nagyjából megegyeznek a C64 készletével, bővítés csupán az ERR, mely a TRAP utasítással kombinálva kiírja a hibás sort.

A fentieken kívül magunk is definiálhatunk tetszőleges számú függvényt (DEF FNxx).

A programozás

A Commodore 16-on BASIC és Assembly nyelvű programok futtathatók, illetve gépi kódú programozásra van lehetőség.

Az első interpreter alatt, az utóbbi kettő a TEDMON, beépített monitor mellett végezhető.

A BASIC +32 767 és −32 768 közötti egész, valamint 10E+38 és 10E−39 közötti lebegőpontos számjegyeket kezel. Duplapontosságú változó használatára és a változótípusok csoportos definiálására nincs lehetőség. Legfeljebb 3 dimenziós numerikus vagy karakterlánctömb használható.

Az OPEN utasításban a fájlok sorszáma 1—255-ig terjedhet. Hogy egyidőben ennyi lánc nyitva lehet-e, arra nincs utalás. A perifériák mindegyike értelemszerűen alkalmazható be/kivitelre.

A kézikönyv

Külön érdemes szólni a géphez mellékelt kézikönyvről. Jóval bővebb és tartalmasabb az eddig megszokottaknál. A gépre vonatkozó rész felépítése a szokásos, ezt egy igen részletes BASIC 3.5 leírás követi, mely önálló BASIC-könyvként is megállná a helyét. A függelék tartalmazza a foglalt kulcsszavakat, a hibakódokat, hibajelzéseket, a nem generált matematikai függvények megadási módját; az egyes frekvenciákhoz tartozó skálahang- és paraméterértékeket a hanggenerátor kezeléséhez; a képernyőkódokat POKE és CHR megadása esetén. Jelentős részt foglal el a TEDMON-monitor kezelésének leírása. A C64-es kézikönyvében találhatónál jóval kevésbé részletezett viszont a memóriatérkép.

A kézikönyv legnagyobb meglepetést keltő része a 3 darab kapcsolási rajz — a hardveresek nagy örömére —, melyből az is kiderül, hogy a gép gyártását 1984. május 29-én engedélyezték. (A hazánkban a Novotrade RT által kiadott magyar nyelvű kézikönyv sem a TEDMON leírását, sem a kapcsolási rajzokat nem tartalmazza. A szerző a Commodore GmbH német nyelvű P/N 324980—02 Nr. 580016 sorszámú kiadványára hivatkozik, amely e tekintetben bővebb. A szerk.)

Értékelés

A gép BASIC-je jobban játékorientált, mint a Commodore 64 BASIC-je: utasításkészlete szinte „aládolgozik" a látványos játékprogramok elkészítésének. (Ugyanakkor a gépi kódú vagy assembly nyelvű programoknál ez már egyáltalán nem igaz, mivel nagyon hiányzik a hardver sprite lehetőség. A szerk.) Sajnos más Commodore gépekre írt programok — éppen az utasításkészlet eltérése és a tárfelosztás különbözősége miatt — általában nem használhatók. Más jellegű alkalmazói szoftverről egyelőre nincs tudomásom, de elképzelhetőnek tartom kisebb méretű, főként grafikus megjelenítést igénylő adatok feldolgozását.

A gép kezelése könnyen elsajátítható, beleértve az újnak számító BASIC-utasítások finomságait is, valamint a TEDMON monitor használatát. Hatékony felhasználásához azonban szükség van a finomságok tökéletes ismeretére.

DEDINSZKY FERENC

Új módszerek a kereskedelemben

A szoftver értékesítése rendhagyó kereskedelmi tevékenység. Következik ez egyrészt a szoftvernek mint árunak néhány sajátosságából, másrészt a hagyományok hiányából. A szoftver ugyanis nagy értékű szellemi újdonságot hordoz, jogvédelem alatt áll, a vételtől várt előny nehezen becsülhető, ezért nagymértékű vásárlói bizalmat feltételez, és a puszta értékesítés is komoly szakértelmet igényel. A kereskedők és fejlesztők fantáziája azonban szinte kimeríthetetlen az értékesítési konstrukciók tekintetében.

A Wirtschafts Woche egyik számában érdekes cikket olvashattunk egy nyugatnémet szoftverfejlesztő kisvállalkozás értékesítési stratégiájáról. A Bad Homburgban működő KHK Source-Software GmbH-ról van szó, amelyet 1983-ban alapítottak, és jelenleg 24 munkatársat foglalkoztat (akárcsak egy magyarországi átlagos kisszövetkezet). Forgalma 1984-ben 3 millió nyugatnémet márka volt, 1985-re 4,2 milliót tervezett. A KHK 60 különböző mikroszámítógép-típushoz gyárt szoftvereket, mintegy tucatnyit géptípusonként.

Hogyan lehet teríteni ezt a sokféle szoftverterméket? Ha egy kereskedő szoftvert szeretne vásárolni a KHK-tól, az ár tízszereséért nemcsak a programcsomagot kapja meg, hanem sokszorosításának jogát is. Ezenfelül kap egy másolóprogramot, amellyel annyi másolatot készíthet a termékről, amennyit csak akar. A KHK szerint jó, ha a kereskedő három példányon túl tud adni. A szoftverek értékesítésében részt vevő partnereknek a forrásprogramot is átadja, ami fokozza a kereskedő biztonságérzetét. Az ügyfelek 50 százalékig terjedő árengedményt is kaphatnak abban az esetben, ha már másodszor jelentkeznek vásárlóként, és más típusú számítógéphez szeretnék beszerezni ugyanazt a szoftvertermékét.

A KHK nem tartja valószínűnek, hogy versenytársai követnék példáját. Szerinte a nagyobb cégeket rabságban tartja a biztonságra való törekvés, és így nem szívesen engedik ki kezükből a sokszorosítás jogát. Nálunk azonban lehetne alkalmazni ezeket a módszereket.

ZS

ADOK — VESZEK — CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,— Ft, magánszemélyeknek az első sor 50,— Ft, minden további sor 20,— Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

● **ZX-SPECTRUMHOZ** játékprogramokat cserélnék. Válaszokat a programok listájával erre a címre kérek: Varga József, Kecskemét, Irinyi út 42. 6000.

● **Új ZX-SPECTRUM 48 K-S** 13 000 forint, 16 k típ. 9000 forint, 2 db botkormány 2500 forint, interfész 3500 forint, 8 db játékkazetta 1200 forintért eladó. Telefon: 76-29 107.

● **VÁMKEZELT COMMODORE 720 -AS** számítógép és 8250 dual floppy eladó. Ottlik András. Telefon: 156-117.

● **1 KBÁJTOS ZX81** magyar nyelvű gépkönyvvel 9000 forintért, valamint 16 kbájtos Commodore 16-os 22 000 forintért eladó. Csontos László, Miskolc, Szervezet u. 30. 3534.

● **C64-ES ÉS DATASETTE** sok programmal eladó 30 000 forintért. Érdeklődni: Béres Antal, Budapest, Labanc út 19. 1021.

● **ZX81+16K RAM** programokkal eladó. Telefon: 49-49-94.

● **ATARI 800XL ÉS 600XL** tulajdonosokat keresek program- és tapasztalatcsere céljából. Lévai Dénes, Budapest, Iglói u. 12. 1118. Telefon: 650-949.

● **AZ INTERBIT ELEKTRONIKAI GMK SZOFTVERAJÁNLATA:**
— Játékprogramok Sinclair ZX81, Spectrum, C64 mikrókra. Kérjen katalógust! Minden ötödik program ingyenes!
— Programfejlesztés: IBM (PC, XT, AT), Commodore 600-700 sorozat, C64, Sinclair Spectrum+, QL stb. PC-kre. Széles körű alapszoftver-kínálat!
— Segédprogramok, tool-ok. Professzionális rendszertámogatás!
— Tanfolyamok. Telephelyen vagy kihelyezett formában: forrásnyelven, magas szintű programnyelveken, gépkezelő- és operátorképzés.
— Referencia, garanciális szolgáltatások, programkarbantartás, hardverbővítés és -illesztés, opciók, helyi hálózatok. Postacím: 1631 Bp. Pf. 11. Részleg: Bp. Bajmóci u. 4. 1112. Telefon: 856-028.

COMMODORE 64

Karakterszerkesztő EPSON nyomtatóhoz

A mindennapi életben egyre gyakrabban találkozunk számítógép által készített táblázatokkal, szövegekkel. Jogos igény a hazai felhasználók részéről, hogy ezek a teljes magyar abc felhasználásával készüljenek. Az ékezetes betűkön kívül sokszor szükséges egyéb speciális jelek, betűk (például matematikai jelek, görög betűk) előállítása. Ezenkívül gyakran előfordul, hogy az alkalmazott számítógép és nyomtató jelkészlete eltérő. Az alábbiakban ezeknek a problémáknak egy lehetséges megoldását ismertetjük a Commodore 64 számítógépen és EPSON FX típusú nyomtatón.

Az *1. ábrán* látható programmal könnyen és gyorsan állítható elő tetszőleges karakterkészlet az EPSON FX típusú nyomtatók számára. Ezzel lehetőség nyílik az ASCII jelkészlet helyett a *Commodore jelkészlet* használatára, így CBM nyomtatóra írt programok működtethetők EPSON nyomtatóval is. A program másik fontos alkalmazási területe a teljes *magyar betűkészlet előállítása.*

A karakterkészlet átdefiniálása nem csökkenti a nyomtatási sebességet, szemben az egyes programok és interfészek által használt grafikus átviteli móddal, amely a nyomtató kapacitását akár a felére is lecsökkentheti. Egy másik előnye ennek a módszernek, hogy nincs szükség a számítógép memóriájára, így bármely meglevő programmal együtt alkalmazható.

Az EPSON EDITOR programmal egy legfeljebb 256 karakterből álló karakterkészlet állítható elő, amely szekvenciális fájl formájában mágneslemezre kerül. A lemezről a karakterkészletet az EPSON LOADER program segítségével lehet betölteni a nyomtató memóriájába.

Néhány szó az EPSON FX típusú mátrixnyomtatók karakterábrázolásáról. Az írásjelek nyomtatása 9 tűvel, 11 pontoszlopban történik. A pontoszlopok távolsága félpontnyi, de a nyomtatott karakterekben ugyancsak a pontsorban a legközelebbi kiírt pontok távolsága egypontnyi. Egy karakter 8 pontsorból áll, a kilencedik (a legfelső vagy a legalsó) pontsor üresen marad. A nyomtató rendelkezik egy „arányos" üzemmóddal, amikor a betűk szélességüknek megfelelően íródnak ki.

Egy írásjel adatai a nyomtatóban 12 bájtot foglalnak el. Az első, az ún. helyzetkód, amely az írásjel függőleges pozicionálásához és az arányos nyomtatáshoz szükséges adatokat tartalmazza, az alábbi felosztás szerint:

7. bit: ha ez a bit 1-es, akkor az írásjel a 9 pontsor közül a felső 8-ba kerül, egyébként az alsó 8-ba
4—6. bit: arányos üzemmódban az első kinyomtatandó pontoszlop sorszáma
Ø—3. bit: arányos üzemmódban az utolsó kinyomtatandó pontoszlop sorszáma.

A helyzetkódot követő 11 bájt a pontoszlopok adatait tartalmazza.

A fentiek alapján a *2. ábrán* látható i betű adatai a következők:
helyzetkód: $151_{10} = 1\ ØØ1\ Ø111_2$
(a betű a felső 8 pontsorba kerül, arányos üzemmódban a 2—7. pontoszlopokat kell kiírni)
a pontoszlopok adatai:
Ø,34,Ø,190,Ø,2,Ø,Ø,Ø,Ø,Ø

```
1 REM ********************************
2 REM *                              *
3 REM *      "EPSON EDITOR 1.2"      *
4 REM *                              *
5 REM ********************************
6 REM
7 REM  NEMCSICS ELEK , 1985. 09. 10.
8 REM
9 GOTO1000
98 :
99 REM"*** ABLAK ***
100 PRINT"■■";
110 PRINT"  ┌─┬─┬─┬─┬─┬─┬─┬─┐ "
111 PRINT"  │ │ │ │ │ │ │ │ │ "
112 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
113 PRINT"  │ │ │ │ │ │ │ │ │ "
114 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
115 PRINT"  │ │ │ │ │ │ │ │ │ "
116 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
117 PRINT"  │ │ │ │ │ │ │ │ │ "
118 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
119 PRINT"  │ │ │ │ │ │ │ │ │ "
120 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
121 PRINT"  │ │ │ │ │ │ │ │ │ "
122 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
123 PRINT"  │ │ │ │ │ │ │ │ │ "
124 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
125 PRINT"  │ │ │ │ │ │ │ │ │ "
126 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┤ "
127 PRINT"  │ │ │ │ │ │ │ │ │ "
128 PRINT"  └─┴─┴─┴─┴─┴─┴─┴─┘ "
130 RETURN
148 :
149 REM"*** MENU ***
150 X$="■■■■■■■■■■■■■■■■"
151 PRINT"■■"X$"■■EPSON EDITOR"
152 PRINTX$"■■0: LEMEZROL OLVASAS"
153 PRINTX$"■1: LEMEZRE IRAS"
155 PRINTX$"■S: SORSZAM ■      ■■■■■■"I
157 PRINTX$"■■+: KOVETKEZO IRASJELHEZ"
158 PRINTX$"■-: ELOZO IRASJELHEZ"
160 PRINTX$"■K: IRASJEL KOD ■      ■■■■■■"K%(I)
161 PRINTX$"■■H: HELYZET KOD ■     ■■■■■■"A%(I)
162 PRINTX$"■■K: PONT BEKAPCSOLAS"
164 PRINTX$"■Z: PONT TORLES"
170 RETURN
198 :
199 REM"*** PONT KEPERNYORE ***
200 PRINTLEFT$(L$,2*K+2+DE)LEFT$(J$,J+2)C$:RETURN
208 :
209 REM"*** KEPERNYOPONT INVERTALAS ***
210 POKEFNS(1),PEEK(FNS(1))OR128:RETURN
211 POKEFNS(1),PEEK(FNS(1))AND127:RETURN
298 :
299 REM"*** IRASJEL KEPERNYORE ***
300 GOSUB1001C$="■■":FORJ=0TO10:FORK=0TO7
302 IFB%(I,J)AND2↑(7-K)THENGOSUB200
310 NEXTK,J:J=0:K=0
320 RETURN
998 :
999 REM"*** START ***
1000 DIMK%(255),A%(255),B%(255,10)
1001 REM" PONT KEPERNYO CIM FUGGVENY
1002 DEFFNS(X)=1024+(2*K+1+DE)*40+J+2
1004 SP$="                                        "
1005 L$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
1006 J$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
1010 POKE53281,0:POKE53280,0
1011 PRINT"■"CHR$(8)CHR$(142);:GOSUB100:GOSUB150
1050 J=0:K=0
1060 DE=2:IFA%(1)AND128THENDE=0
1061 GOSUB150:GOSUB100:GOSUB300:GOSUB210
1100 GETA$:PRINT"■";
1102 IFA$="0"THEN1200
1103 IFA$="1"THEN1300
1104 IFA$="S"THEN1400
1105 IFA$="+"THEN1500
1106 IFA$="-"THEN1600
1107 IFA$="K"THEN1700
1108 IFA$="H"THEN1800
1109 IFA$="X"ORA$="Z"ORA$="*"THEN1900
1110 IFA$="Z"ORA$="*"THEN1950
1111 IFA$="■"THENGOSUB211:J=J-(J<10):GOSUB210
1112 IFA$="■"THENGOSUB211:J=J+(J>0):GOSUB210
1113 IFA$="□"THENGOSUB211:K=K+(K>0):GOSUB210
1114 IFA$="■"THENGOSUB211:K=K-(K<7):GOSUB210
1120 GOTO1100
1198 :
1199 REM"*** JELKESZLET BEOLVASAS ***
```

1. ábra

(Folytatása a 31. oldalon.)

```
1200 PRINTLEFT$(L$,22);:A$="":INPUT"■LEMEZROL■
     ADATOK NEVE";A$:PRINT"□"SP$
1203 IFA$=""THEN1060
1204 IFLEN(A$)>16THEN1200
1205 OPEN1,8,15,"I"
1206 OPEN8,8,8,A$+",S,R"
1208 I=0
1210 PRINTI"□":INPUT#8,K%(I),A%(I)
1220 FORJ=0TO10:INPUT#8,B%(I,J):NEXT
1222 IFI<255ANDA%(I)<>0THENI=I+1:GOTO1210
1230 CLOSE8:CLOSE1:PRINTSP$:GOTO1060
1298 :
1299 REM"*** JELKESZLET ELMENTESE ***
1300 PRINTLEFT$(L$,22);:A$="":INPUT"■LEMEZRE■
     ADATOK NEVE";A$:PRINT"□"SP$
1303 IFA$=""THEN1060
1304 IFLEN(A$)>16THEN1300
1305 OPEN1,8,15,"I"
1306 OPEN8,8,8,"@0:"+A$+",S,W"
1308 I=0
1310 PRINTI"□":PRINT#8,K%(I):PRINT#8,A%(I)
1320 FORJ=0TO10:PRINT#8,B%(I,J):NEXT
1322 IFI<255ANDA%(I)<>0THENI=I+1:GOTO1310
1330 CLOSE8:CLOSE1:PRINTSP$:GOTO1060
1398 :
1399 REM"*** MASIK IRASJELHEZ ***
1400 PRINTLEFT$(L$,22);:INPUT"SORSZAM";I
1402 I=INT(I):PRINT"□"SP$:IFI>=0ANDI<256THEN1060
1405 GOTO1400
1498 :
1499 REM"*** KOVETKEZO IRASJELHEZ ***
1500 I=I-(I<255):GOTO1060
1598 :
1599 REM"*** ELOZO IRASJELHEZ ***
1600 I=I+(I>0):GOTO1060
1698 :
1699 REM"*** IRASJEL KODJANAK MEGADASA ***
1700 PRINTLEFT$(L$,22);:INPUT"KOD";K%(I):PRINT"□"SP$
1704 IFK%(I)>=0ANDK%(I)<256THEN1060
1705 GOTO1700
1798 :
1799 REM"*** HELYZET KOD MEGADASA ***
1800 PRINTLEFT$(L$,22);:INPUT"HELYZET KOD";A%(I)
     :PRINT"□"SP$
1804 IFA%(I)>=0ANDA%(I)<256THEN1060
1805 GOTO1800
1898 :
1899 REM"*** PONT BEKAPCSOLASA ***
1900 C$="■■■":GOSUB200:B%(I,J)=B%(I,J)OR2↑(7-K)
     :GOTO1100
1948 :
1949 REM"*** PONT TORLESE ***
1950 C$="■■ ":IFJ=1ORJ=3ORJ=5ORJ=7ORJ=9THENC$="■■|"
1952 GOSUB200:B%(I,J)=B%(I,J)AND(255-2↑(7-K)):GOTO1100
```

Az 1. ábra folytatása

Az EPSON EDITOR program beolvasása után a képernyőn az i betű képe látható, a jobb oldalon megjelennek a programban használható parancsok és az éppen szerkesztett írásjel egyes adatai. A kívánt írásjelet a kurzor mozgatásával és az X, illetve a Z gombbal lehet megrajzolni. Az X és Z parancsok a kurzorvezérlő gombok folyamatos lenyomva tartása mellett is működnek, ilyenkor a pont be- vagy kikapcsolása után a kurzor továbblép a kívánt irányba. Ezután megadjuk a helyzetkódot a H gombbal és a karakterkódot a K gombbal.

Ezután áttérhetünk a következő karakter szerkesztésére vagy javítására. A + parancs hatására az előzőnél eggyel nagyobb, a − parancsra pedig az eggyel kisebb sorszámú írásjel adatai jelennek meg. Lehetőség van közvetlenül egy kívánt sorszámú írásjel előhívására is az S gombbal. A különböző írásjelek megadását a 2. ábrán néhány példa illusztrálja.

Az így megszerkesztett jelkészletet az I paranccsal mágneslemezre kell rögzíteni. A felvétel az első nulla helyzetkódú karakterig történik. Ez a karakterkészlet javítás és bővítés céljából az O paranccsal a lemezről vissza is olvasható. Az ADAT NÉV? kérdésre csak a RETURN gombbal válaszolva a tévedésből kiadott I és O parancs hatástalanítható.

Néhány kiegészítés az EPSON LOADER programot tartalmazó *3. ábrához*. A fontosabb változók definíciója:

k%(i): az írásjelkódok vektora
a%(i): a helyzetkódok vektora
b%(i,j): a pontoszlopadatok
i: a szerkesztett jel sorszáma
j: a pillanatnyi pontoszlop sorszáma
k: a pillanatnyi pontsor sorszáma

2. ábra

```
1 REM *********************************
2 REM *                               *
3 REM *       "EPSON LOADER 1.0"      *
4 REM *                               *
5 REM *********************************
6 REM
7 REM  NEMCSICS ELEK , 1985. 09. 12.
8 REM
10 DIM8(10)
11 OPEN2,4,4
12 N$=CHR$(0)
13 :!REM  KARAKTERGENERATOR 'RAM'-BA MASOLASA
14 PRINT#2,"■■"N$N$N$N$;
15 :!REM  'RAM'-BELI KARAKTERKESZLET AKTIVIZALASA
16 PRINT#2,"■■"CHR$(1)N$;
19 PRINT"A KARAKTERKESZLET NEVE:":INPUTXX$
20 OPEN1,8,15:OPEN8,8,8,XX$
29 :!REM  'KOD' ES 'HELYZET KOD' BEOLVASASA
30 INPUT#8,K,A
39 :!REM  PONTOSZLOP ADATOK BEOLVASASA
40 FORJ=0TO10:INPUT#8,B(J):NEXT
49 :!REM  HA A 'HELYZET KOD 0, VEGE
50 IFA=0THENCLOSE8:CLOSE1:CLOSE2:END
59 :!REM  A 'K' KODU 'RAM'-BELI IRASJEL ATDEFINIALASA
70 PRINT#2,"■■"N$CHR$(K)CHR$(K)CHR$(A);
80 FORJ=0TO10:PRINT#2,CHR$(B(J));:NEXT
90 GOTO30
```

3. ábra

```
1 REM *********************************
2 REM *                               *
3 REM *     "EPSON TESTER 1.0"        *
4 REM *                               *
5 REM *********************************
6 REM
7 REM  NEMCSICS ELEK , 1985. 09. 11.
8 REM
10 OPEN4,4,4
20 FORI=0TO255:A$=CHR$(I):IF (I<32) OR ((I)128)
   AND (I<160)) THEN A$="."
30 PRINT#4,RIGHT$("    "+STR$(I),6)+" "+A$
40 IF(I+1)/8 =INT((I+1)/8 )THENPRINT#4
50 NEXT
60 PRINT#4:CLOSE4
```

4. ábra

A Commodore 64-ben a logikai kifejezések értéke „igaz" esetben −1, egyébként 0. Ezért a program más gépre való adaptálásánál ügyeljünk az aritmetikai kifejezésekbe ágyazott logikai kifejezések előjelére. A program a STOP gombbal leállítható. Folytatás a CONT vagy a GOTO1060 utasításokkal lehetséges.

A lemezen levő jelkészletet az EPSON LOADER segédprogrammal vihetjük be a nyomtatóba. A beolvasott jelkészletben nem szereplő kódokhoz továbbra is a nyomtató eredeti írásjelei tartoznak. A nyomtató éppen érvényes jelkészletét a *4. ábrán* látható programmal lehet kiíratni.

NEMCSICS ELEK

COMMODORE 64

Megszakítás II.

A VIC (video interface chip) megszakítása hasznos lehet 8-nál több sprite létrehozásánál vagy a border színesebbé tételénél. Ez utóbbin keresztül mutatom be a megszakítás eredményes használatát.

A képernyő 300 sorból áll. Ebből 200 a látható képernyő, mely az 51. sortól kezdődik. A képernyő szerkesztésekor a VIC 18. regisztere ($D012(53266)) tájékoztat az aktuális rasztersorról. Mivel egy bájton csak 256 az ábrázolási lehetőség, ezért szükség van még egy bitre. Ez a VIC 17. regiszterének ($D011(53265)) 7. bitje. Ha ezt a megszakítási módszert akarjuk használni, akkor először szükség van a raszter beállítására, majd a megszakítás engedélyezésére. Az engedélyezést a VIC 26. regisztere ($D01A(53274)) végzi. Az első bit a rasztere. Ha ez magas, akkor történik megszakítás. A bájt képe a következő:

7	6	5	4	3	2	1	0
–	–	–	–	ELP	EMMC	EMDC	ERST

A raszter beállítását ugyanazok a bitek végzik, amelyek jelzik az aktuális sorszámot. Az ezekbe beírt értéket a gép elteszi magának, és ha az elmentett érték megegyezik az aktuálissal, akkor megszakítás generálódik. Ennek lekérdezésére a VIC 25. regisztere ($D019(53273)) szolgál. Bitjei megfelelnek az engedélyező regiszter bitjeinek, azzal a módosítással, hogy a megszakításkor a megfelelő bit 1-es lesz. Ennek törlését a regiszter visszaírásával érhetjük el. A bájt képe ilyen lesz:

7	6	5	4	3	2	1	0
IRQ	–	–	–	ILP	IMMC	IMDC	IRST

A rasztermegszakításnál érdemes mindenféle megszakító rutint magunknak kezelnünk, mert előfordulhat, hogy a megszakítások egymásba érnek, így lemerevedhet a gép. Példánknál az is megeshet, hogy az átmeneteknél csúnyán vibrálni fog a kép. Ugyancsak vibrálás az eredmény, ha annak megállapítására, hogy melyik sornál történt megszakítás, nem egy külön bájtot használunk, hanem a raszterregisztert.

A programban erre a HOL bájt szolgál. A SZIN a színek, a SOR pedig azoknak a soroknak az értékét tartalmazza, ahol a megszakítást szeretnénk. A megszakító rutinban a szín és a következő raszterérték beállítása történik.

MÜLLER GÁBOR

```
RASTER.A.....PAGE 0001

LINE# LOC   CODE        LINE


00001  0000                    #=$7000
00002  7000  78                SEI
00003  7001  AD 14 03          LDA $0314       ;IRQ VEKTOR REGI CIMENEK ELMENTE
SE
00004  7004  8D 92 70          STA VISSZA+1    ;AZ UJ BEIRASA
00005  7007  A9 37             LDA #<IRQ
00006  7009  8D 14 03          STA $0314
00007  700C  AD 15 03          LDA $0315
00008  700F  8D 93 70          STA VISSZA+2
00009  7012  A9 70             LDA #>IRQ
00010  7014  8D 15 03          STA $0315
00011  7017  AD A2 70          LDA SZIN+2      ;BORDER SZIN BEALLITASA
00012  701A  8D 20 D0          STA $D020
00013  701D  AD 11 D0          LDA $D011       ;RASZTER 8.BITJENEK TORLESE
00014  7020  29 7F             AND #$7F
00015  7022  8D 11 D0          STA $D011
00016  7025  AD A5 70          LDA SOR+2       ;RASZTER SOR BEALLITASA
00017  7028  8D 12 D0          STA $D012
00018  702B  A9 7F             LDA #$7F        ;CIA-CHIP MEGSZAKITAS LETILTASA
00019  702D  8D 0D DC          STA $DC0D
00020  7030  A9 01             LDA #$01        ;RASZTER MEGSZAKITAS ENGEDELYEZE
SE
00021  7032  8D 1A D0          STA $D01A
00022  7035  58                CLI
00023  7036  60                RTS
00024  7037                ;
00025  7037                ; MEGSZAKITO RUTIN
00026  7037                ;
00027  7037  AD 19 D0    IRQ   LDA $D019       ;RASZTER MEGSZAKITAS TORTENT-E
00028  703A  29 01             AND #$01
00029  703C  D0 03             BNE EGY
00030  703E  4C 9A 70          JMP EOF
00031  7041  AD A6 70    EGY   LDA HOL
00032  7044  C9 00             CMP #$00
00033  7046  D0 12             BNE KETTO
00034  7048  AD A3 70          LDA SOR
00035  704B  8D 12 D0          STA $D012
00036  704E  AD A0 70          LDA SZIN
00037  7051  8D 20 D0          STA $D020
00038  7054  EE A6 70          INC HOL
00039  7057  4C 94 70          JMP KI
00040  705A  AD A6 70    KETTO LDA HOL
00041  705D  C9 01             CMP #$01
00042  705F  D0 12             BNE HAROM
00043  7061  AD A4 70          LDA SOR+1
00044  7064  8D 12 D0          STA $D012
00045  7067  AD A1 70          LDA SZIN+1
00046  706A  8D 20 D0          STA $D020
00047  706D  EE A6 70          INC HOL
00048  7070  4C 94 70          JMP KI
00049  7073  AD A6 70    HAROM LDA HOL
00050  7076  C9 02             CMP #$02
00051  7078  D0 28             BNE EOF
00052  707A  AD A5 70          LDA SOR+2
00053  707D  8D 12 D0          STA $D012
00054  7080  AD A2 70          LDA SZIN+2
00055  7083  8D 20 D0          STA $D020
00056  7086  AD 19 D0          LDA $D019       ;MEGSZAKITAS TORLESE
00057  7089  8D 19 D0          STA $D019
00058  708C  A9 00             LDA #$00
00059  708E  8D A6 70          STA HOL
00060  7091  4C FF FF    VISSZA JMP $FFFF      ;EREDETI MEGSZAKITO RUTIN VEGREH
AJTASA
00061  7094  AD 19 D0    KI    LDA $D019       ;MEGSZAKITAS TORLESE
00062  7097  8D 19 D0          STA $D019
00063  709A  68          EOF   PLA             ;REGISZTEREK VISSZATOLTESE
00064  709B  A8                TAY
00065  709C  68                PLA
00066  709D  AA                TAX
00067  709E  68                PLA
00068  709F  40                RTI
00069  70A0  02          SZIN  .BYTE $02,$01,$00
00069  70A1  01
00069  70A2  00
00070  70A3  64          SOR   .BYTE $64,$C8,$00
00070  70A4  C8
00070  70A5  00
00071  70A6  00          HOL   .BYTE $00
00072  70A7                    .END

ERRORS = 00000


SYMBOL TABLE

SYMBOL VALUE
 EGY     7041   EOF     709A   HAROM   7073   HOL     70A6
 IRQ     7037   KETTO   705A   KI      7094   SOR     70A3
 SZIN    70A0   VISSZA  7091

END OF ASSEMBLY
```

COMMODORE

Módosítások Commodore gépeken

Néhány jól használható, egyszerűen egyetlen címen levő tártartalom megváltoztatásával megvalósítható módosítást imertetünk a táblázatban, a Computer Gazette című lap nyomán. A táblázat a legelterjedtebb négy Commodore gépre adja meg a módosítandó címet és az új tartalmat (a Plus 4 és a C16 címei azonosak).

HATÁS	C–64	VIC–20	C–16, PLUS 4
LIST hatástalanító	775,191	775,223	774,187
LIST engedélyező	775,167	775,199	774,110
SAVE hatástalanító	819,246	818,73	816,136
Engedélyező	819,245	818,133	816,164
LOAD hatástalanító	816,157	816,103	814,239
Engedélyező	816,165	816,73	814,74
RUN/STOP hatástalanító	808,239	808,100	806,103
Engedélyező	808,237	808,112	806,101
RESTORE hatástalanító	792,193	792,7	—
Engedélyező	792,71	792,173	—
Billentyűzethatástalanító	649,0	649,0	1343,0
Engedélyező	649,10	649,10	1343,10
Billentyűzetismétlés-tiltó	650,64	650,64	1344,64
Engedélyező	650,128	650,128	1344,128
Csak a léptető, a DEL és a CRSR ismételhet	650,0	650,0	1344,0
Billentyűzetpuffer-törlő	198,0	198,0	239,0
Fénypontmozgatási sebességbeállító (0–255, a 0 a leggyorsabb)	56325,x	37159,x	—
Karakterszin-beállítás (VIC–20-nál 0–7, a többinél 0–15)	646,x	646,x	1339,x
Sorszámkiiratás letiltása LIST-nél	22,35	22,35	22,35
Engedélyezés	22,25	22,25	22,25

Figyelem! Ha a billentyűzet használatát letiltjuk, ezt a POKE-pár bebillentyűzésével nem lehet feloldani, mert azt nem tudjuk beadni! Ilyenkor először a RUN/STOP-RESTORE billentyűpárt kell használni.

Óvatosan bánjunk a listázás letiltásával! Előfordulhat, hogy hibás a kiírás.

Egy kissé bonyolultabb, de nagyon fontos módosítás, amivel a tévesen beadott NEW parancs hatástalanítható:

C64

```
POKE2050,1:SYS42291:POKE45,PEEK(34):POKE46,
PEEK(35):CLR
```

VIC–20

```
POKE4098,1:SYS5048 3:POKE45,PEEK(34):POKE46,
PEEK(35):CLR
```

Ez utóbbi gép 8 k vagy nagyobb bővítésű változatainál az első cím 4610.

PRIMO

Príma tippek

1. Ha le akarjuk állítani a programot, hasznunkra lehet a billentyűzet letiltása. Talán nem sok Primo-felhasználó tudja, hogy POKE 16408,1 hatására a billentyűzet „megnémul", és ezen már csak a RESET gomb segít. A billentyűzet engedélyezése: POKE 16408,195

2. A képernyő gyors befestésére több eljárást ismerünk, de az egyik lassú, a másik gépi kódú, amit nem tud még mindenki, a harmadik pedig nem valódi befehérítés (CLS után egy PRINT CHR$(3)). Az én módszerem viszonylag gyors és egyszerű.

```
10 CLS
20 FOR I=0 TO 255
30 A=CALL(16060),I)
40 NEXT I
```

Ha a 30-as sorban az I helyett –I-t (mínusz I-t) írunk, akkor fordított irányban történik a befestés. Két irányból is megoldható a feladat:

```
20 FOR I=0 TO 128
25 A=CALL(16060),255–I)
```

A többi sor megegyezik.

3. A HT géphez hasonlóan a Primón is vannak nem definiált utasítások, melyeket a 16722-es címtől a 1680A-es címig találunk a memóriában. Ezek közül azonban nem mindegyiket lehet átírni. Az átírható utasítások a 16761-es címtől szerepelnek. Néhány gyakrabban átírt utasítás:

Utasítás	*Cím*
FIELD	16764
GET	16767
PUT	16770
MERGE	16779
KILL	16785
LINE	1680A (ez az utolsó az összes közül)

4. Talán kevesen tudják, hogy még néhány definiált utasítást (RUN, LIST, GOTO stb.) is „meg lehet lőni", át lehet írni. Ez BASIC-ből POKE-okkal valósítható meg. Az ad rá lehetőséget, hogy ezek az átírható utasítások végrehajtásuk során „kiugranak" a RAM-ba egy meghatározott címre. Itt, ezen a címen alapállapotban egy C9(RET) van, amelyet ha átírunk, tetszőleges feladatot adhatunk egy-egy utasításnak. Például

LIST letiltása:

```
POKE 16863,195,204,6
```

Csak GOTO indítás:

```
POKE 16827,195,204,6
```

Ez tulajdonképpen a BASIC-be ugrás (JP 06CC).

5. Az előbbi pontban leírtakat használja ki a következő, kissé furcsa eredménnyel járó trükk. A programrészlet begépelése után két példaprogramot kell beírni. A RUN parancsra elindul az egyik program. Ennek vége után, újabb RUN parancsra már a másik program indul el, és ez így váltakozik a RUN parancsokra. A program a következő:

```
 5 Z=PEEK(16629)+PEEK(16630)*256
 7 IF Z=0 OR Z=400 THEN 10 ELSE 200
10 INPUT N
20 FOR I=1 TO N
30 SET(I,I–50)
40 NEXT I:STOP
200 INPUT N
210 A=SIN(n)+COS(N)
220 PRINT "EREDMÉNY=";A
400 PRINT "VÉGE"
```

NÓGRÁDI ELŐD

Karterminál

A japán Seiko cég karóra formájú és méretű számítógépterminált hozott forgalomba. Ez megfelelő illesztő segítségével ma már minden elterjedtebb személyi számítógéphez csatolható, és abból e terminálba információk, kisebb adatállományok tölthetők. Így például a megfelelő időpont közeledtére a karterminál csipogással figyelmeztet, vagy megnézhetjük rajta a soron következő tárgyalásra vonatkozó legfontosabb feljegyzéseket.

A HIBA NEM AZ ÖN KÉSZÜLÉKÉBEN VAN

A µM 1985/3. számában megjelent Ki ad magyarázatot? című cikk az 1984/6. szám termékismertetőjében felsorolt Primo „bug"-okra vár magyarázatot.

Úgy vélem, a tényleges vagy vélt hibák ismertetése helyett célravezetőbb lenne a meglevő gép felhasználására adni tanácsot, hogy a nem kívánt „furcsaságokat" elkerüljük vagy a magunk hasznára fordítsuk.

A géphez adott Felhasználói kézikönyv bizony valóban kevés támogatást ad azoknak a gyakorlottabb programozóknak, akik a gépben rejlő lehetőségeket szeretnék minél jobban kiaknázni. Az ismerkedés így különféle segédprogramocskákkal lehetséges csak, de itt és most is szeretném felhívni a figyelmet arra a „sajnálatos" tényre, hogy a Primo is SZÁMÍTÓGÉP (de hogy mikro, személyi, „nép-" vagy hobbi-e, azt nem vagyok hivatott eldönteni), s mint ilyen, a nem elhanyagolható számítástechnikai és matematikai ismereteken kívül figyelmet is követel a programok írása során. Ráadásul a Primo az egyszerűbb hardver mellett inkább bonyolultabb szoftverrel biztosítja a sokoldalú felhasználhatóságot, tehát az átlagosnál talán nagyobb figyelmet kíván. „Vad" programok, melyek végigpróbálgatják a vezérlőkódokat (lásd az ismertető 3. képét), sőt össze-vissza módosítják a kommunikációs terület lényeges információit (lásd 4. kép), természetesen megvadíthatják a gépet is, hiszen nem „fool-proof"-nak készült.

Engedtessék meg nekem, hogy hely hiányában a mindenre kiterjedő és korrekt magyarázatok helyett csak mazsolázgassak a gépnek az „ismertető"-ben olvasható „kétségtelenül szép számban fellelhető »bogarai«-ból".

A 152—255 ASCII-kód tartománynak semmiféle *értelmes* grafikus kódmegfeleltetése nincs mindaddig, amíg nem adunk erre vonatkozóan információkat a gépnek.

A Primo ugyanis nem a megszokottabb videoprocesszorral és karaktergenerátorral működik, hanem külön szubrutin viszi át a karaktermintázatot a RAM utolsó 6 kbájtnyi területére, melyet speciális hardver visz ki a képernyőre. Ez ugyan lassítja valamelyest a működést, de rendkívüli rugalmasságot biztosít.

A karakterek mintázatát rögzítő tábla a ROM 31F7H — 35C6H területén helyezkedik el. Ez a 3Ø—151 kódú karakterek generálásához biztosítja a szükséges adatokat. A 128—255 kódú karakterek azonban szabadon is definiálhatók, mivel a 128 és efeletti kódú karakterek mintázatát rögzítő tábla kezdőcíme — mely bekapcsolás, illetve RESET után 35Ø7H —, a 4Ø4BH—4Ø4CH című rekeszekben található, és így átírható. Ha az ide beírt kezdőcímtől kezdődően elhelyezzük a kívánt mintázatot, akkor a gép a következő RESET-ig ezeket fogja a képernyőre vinni. RESET után ezt az új kezdőcímet ismét be kell írni!

A PRINT utasítás hatására egy karakterhez általában 8 bájtot olvas ki és ír át a képernyőmemória kurrens részébe a 4Ø48H és 4Ø49H-ban tárolt X és Y koordináták alapján, de a kiolvasás valójában addig tart, amíg a mintázatbájtok Ø. bitje át nem vált 1-re. Így a karakterek tetszőleges függőleges méretűek lehetnek, de azon az áron, hogy így több karakterkódot is lefoglalnak, mivel a mintázat első (legfelső) bájtja a kar.tábla kezdőcím+(kar.kód−128)*8 címen helyezkedik el. A karaktermintázatot rögzítő bájtok a 6. . .1 biteken tárolják a pontinformációt (az 1. bit általában betűköz, tehát Ø). A 7. bit csak a legelső bájtnál számít: ha 1, akkor ez ún. „süllyesztett" karakter (g, j, p, q, y) lesz, és 2 pontsorral lejjebb kezdődik a beírása, a Ø. bit pedig 1-es értékével az utolsó kiolvasandó bájtot definiálja.

Az alábbi kis program a hosszított integráljel kiírását teszi példaként lehetővé a 128 kóddal:

```
1Ø POKE16459,Ø,96 'új kar.tábla cím:
   24576=6ØØØH
2Ø POKE24576,Ø,12,18,16,16,24,24,24
   24,24,24,24,8,8,72,49
3Ø CLS:?:?"+3":?CHR$(17)"
   "CHR$(128) CHR$(16)"5xdx=";
   (5*3!2)/2-(5*(-8)!2)/2:?"-8":?
```

A rendelkezésre álló Primo memóriaméretétől függően azonban szükséges lehet a RAM elejének vagy végének „lefoglalása", hogy azt a BASIC-változók ne írhassák át. A 4ØA4H—4ØA5H tartalmazza a BASIC programok mindenkori kezdőcímét. Ezt átírva a tár elejéből kapunk szabad területet. Az átírás után azonban ki kell adni a NEW parancsot, hogy a gép megfelelően átírja a rendszerváltozókat.

A 4ØB1H—4ØB2H átírásával viszont a memória végéből kaphatunk szabadon felhasználható területet, utána a CLEAR5Ø parancsot kell kiadni a rendszerváltozók átírása érdekében.

A kétszeres pontosság a valóságban a számítás pontossága is, nem csupán a kiírásé, mely az elemi műveletek esetén jól megfigyelhető. A gép azonban a függvények jelentős részénél különféle *közelítéssel* dolgozik, s az így keletkező hibák felhalmozódása után valóban megtéveszthető „dupla pontossággal" írja ki az utolsó néhány pontostól eltérő számjegyét is. A közelítések lehetnének pontosabbak, de ez a számítások idejét jelentősen megnövelné, s a tervezők ezt a kompromisszumos megoldást választották.

Valójában ritkán is van ténylegesen szükség ilyen nagy pontosságra, s ekkor az elemi műveletekkel kell felépíteni a szükséges függvényeket is.

A logikai IGAZ értéke valóban −1, de ennek „szokatlan" volta kérdéses (és *szokás* dolga), mivel így dolgozik — többek között — a VIC20, C64, HT—1080Z gép is. Oka egyszerű. A két bájton tárolt logikai IGAZ bináris megfelelője 11111111 11111111, míg a logikai HAMIS-é ØØØØØØØØ ØØØØØØØØ, tehát akkor IGAZ, ha minden bit IGAZ, és akkor HAMIS, ha minden bit HAMIS. A csupa 1-es azonban éppen a decimális −1 megfelelője.

A Primo (és jó néhány más gép is) gyakran azt az értéket tekinti IGAZ-nak, ami nem HAMIS, vagyis nem Ø. Például:

```
1Ø A=2Ø:B=Ø
2Ø IFA=2ØTHENC=5
3Ø IFATHENC=5
4Ø IFB=ØTHENC=5
5Ø IFNOT(B)THENC=5
6Ø IFBELSEC=5
```

a 2Ø és 3Ø, illetve a 4Ø. . .6Ø sorok azonos eredményt adnak, C értéke mindig 5 lesz.

Az ON N GOTO . . . utasításnál az N=Ø és N lista tételszám esetén való továbblépése (és nem hibajelzéssel való leállása) sem teljesen szokatlan. Ráadásul nagyon jól ki is használható ez a több gépen kitűnően bevált megoldás, mivel csak arra kell igazán ügyelni, hogy N értéke negatív ne lehessen. A következő sor például „javíthatja" a nem megfelelő N értékét.

Az ismertető 1. képén látható kis- és nagybetűk keveredése valóban többé-kevésbé egyedi jelenség. A túlságosan érzékenyre állított billentyűzet a SHIFT-et már a kéz közelítésekor is megérintettnek érzékeli, s így áll elő a „szemmelverés" effektusa. A gépben levő pontenciométerrel ez beállítható, de ez egyelőre csak a kijelölt szervizekben ajánlatos a garanciális következmények miatt. Egyszeri jó beállítás után már ritkán okoz problémát.

Az ismertető 2. képe azt igyekszik bizonyítani, hogy a karakterek helyfoglalása a memóriában nem számítható előre. Bár a bizonyítást nem értem egészen, de tény, hogy egy karakter a Primóban is egyetlen bájt helyet igényel csupán. Természetesen szükség van a változó azonosítására, típusának, verembeli címének, a karakterek számának és esetleg a dimenzió(k) rögzítésére is, melyek a változótáblán foglalnak helyet. Vegyük például a következő programot:

```
10 CLEAR50:FORI=1TO10:?FRE(X$),
   :A$=A$+"!":?LEN(A$):NEXT
```

Jól látható futtatás után a stringek számára lefoglalt 50 bájtnyi terület szabad részének csökkenése a string hosszának növekedésével.

Az ismertető 3. képén valóban furcsa a lista. Ez azonban a kellő megfontolás nélkül végrehajtott vezérlőkódoknak köszönhető. A karakterek csonkulását az okozta, hogy a függőleges írásra váltás (ez az ábrán nem látható már), majd vízszintesre visszaváltás után nem határozta meg a következő sor pozícióját, s így a gép csak az utolsó karakter alapján határozza meg az Y koordinátát. Ha ekkor a 16457 (4049H) című rekesz tartalmának 1-gyel csökkentett értéke nem osztható maradék nélkül 12-vel, akkor a 15. sor írása utáni automatikus scroll során a 12 pontos karaktersorból akár 11 is elveszhet, s csak a nagybetűk ékezetének felső pontja látható. Ez a jelenség természetesen kivédhető lenne a gép szoftverével is, de akkor a grafikus lehetőségek csökkennének, s így nem látszik célszerűnek. A minden eshetőségre kiterjedő szabályok felsorolása helyett az alábbiak betartásával, illetve elvégzésével a karaktercsonkulások elkerülhetők:

1. Ne próbáljunk a legfelső sorban felső indexet használni, mert a gép azt úgysem veheti figyelembe.

2. Egy soron belül az alsó- és felsőindex-utasítások száma lehetőleg egyezzen meg, de különbségük mindenképpen legyen maradék nélkül osztható 3-mal (figyelembe véve az 1. pontban írtakat is).

3. Függőleges írásmódból történő visszaváltás után a következők közül célszerű választani:

a) A következő kiíratható utasítás PRINT$y,x, . . . legyen.
b) Iktassuk be a következő programsort:
   POKE16457,INT((PEEK(16457)−1)/12)*12+1
c) CLS utasítás.

Az ismertető 4. képéről nincs sok mondanivaló. Pregnáns példája az „így-ne-programozzunk" esetének. Mit is csinál ez a program? Átír minden olyan rekeszt, amely a 16384. . .17000 területen 0 vagy 255 értékű (a 16383-at nem tudja átírni, mert az még ROM).

Mivel a Primo éppen ezen a területen tárolja a különféle RST és egyéb ugróutasításait, valamint a rendszerváltozókat, így nincs mit csodálkozni a gép „elszállásán". (Csak zárójelben, hogy az NMI vezeték ilyen módon aligha kaphat jelet, viszont a Primónál e jel RESET hatására történő aktivizálása letiltható a 403BH 7. bitjének 0-ra történő átírásával, s ilyenkor a RESET gomb hatástalanná válik.)

A billentyűismétlés ízlés dolga. Nekem a 3,75 MHz-cel működő rendszerben (mert legtöbb gép erre is átállítható!) sem okozott soha problémát. Akinek ez mégis gyors, lehetősége van a lassításra (de gyorsításra is), amennyiben a 4046H rekeszben található 24-es értéket nagyobbra átírja (10. . .255 között kell, hogy legyen).

Összegezve: a rendelkezésemre álló 48 k-s gép március óta az időnkénti *kikapcsolás* mellett kifogástalanul üzemel mind a BASIC nyelvű, mind az általam adaptált assemblerrel írt gépi nyelvű programokkal.

GAÁL TAMÁS
Salgótarján, Ybl Miklós út 88.

A rovatszerkesztő megjegyzései

Úgy tapasztaltam, ha valaki azt írja, hogy „a . . . hibák ismertetése helyett" csináljunk valami ún. építő jellegű dolgot, például dicsérjük a gép előnyeit, az nem tűri a bírálatot. Én viszont a Primóról igyekeztem elfogulatlan cikket írni. Egy ilyen cikk természetesen nem állhat csak dicséretből.

Ezután nézzük a megjegyzéseket!

1. A levéliró hibáztat, mert nem tudva azt, hogy a Primo „egyszerűbb hardver mellett inkább bonyolultabb szoftverrel biztosítja a sokoldalú felhasználhatóságot", olyan „vad" programokkal vizsgáltam a memóriát, melyek különböző módosításokat okoztak a gép normális üzemmenetében. Kénytelen vagyok visszakérdezni: Mindezt tudnom kellett volna a felhasználói kézikönyv alapján? Más információ ugyanis sem nekem, sem az akkori vásárlónak nem állt rendelkezésre. Úgy vélem, és ebben aligha tévedek, hogy a gépet rendeltetésszerűen használtam, és ha ilyenkor valami olyasmi történik, ami a normális üzemet zavarja, és amelynek tilalmára a figyelmet nem hívták külön fel, az a gyártó hibája és nem a felhasználóé.

2. A grafikus karakterek előállításmódját az olvasó honnan ismerte meg? A felhasználói kézikönyvből? Végigpróbálta a memóriát a 31F7H eléréséig valamilyen nem „vad" programmal? Ugyanezt kérdezhetem a kiíratás formátumával kapcsolatban is. A válasz nyilvánvalóan nem. Eredeti cikkemben leírtam, hogy a felhasználói kézikönyv nem elégséges. Ezzel a gyártó is egyetértett, és azóta hardver és szoftver segédkönyvet jelentetett meg.

3. Honnan tudja az olvasó azt, hogy az ún. kétszeres pontosság előállítására milyen módszert használtak a tervezők? Vele külön közölték vagy ő is a tervezők egyike? Függetlenül ettől, ha nincs leírva, hogy a kétszeres pontosság csak az alapműveletekre értendő, úgy annak igaznak kell lennie a négyzetre emelésre és a hatványozásra is. Egyébként a szokásos transcendens függvényeknél — a logaritmus kivételével — a kétszeres pontosság elérése csak jelentéktelen mértékben növeli az előállításához szükséges gépi időt. Leírni azt, hogy egy BASIC-fordító kétszeres pontosságú (ilyen korlátozások mellett, a korlátozások említése nélkül), megtévesztő. A gyártó mentségére persze felhozható, hogy ugyanezt tette korábban a Híradástechnika Szövetkezet is a HT—1080Z gépnél!

4. A 2. képről hiányzik a program első három sora:

```
 5 CLEAR500:A$=""
10 FORI=23TO255
20 A$=A$+CHR$(I)+"I="
   +STR$(I)
```

A program tehát növekvő hosszúságú karaktersorozatot állít elő, míg hibajellel le nem áll. A hibajelzés i=51 értéknél volt. Ekkor a karaktersorozat 168 elemből áll. Betoldva a 25 B$=A$ sort, és a 40 sorhoz hozzáírjuk LEN(B$) szöveget. A hibajelzés most J=44 esetén következik be, amelynél mindkét karaktersorozat hossza 126 elem.

Más gépeknél, néhány bájt eltéréssel kiszámítható a változók (numerikus, karakteres, tömb) helyfoglalása a következő képlettel:

A*N+B

Itt A, B a változó típusára jellemző állandó, ami természetesen azonos változótípusra azonos. N a számjegyek, karakterek száma. Úgy vélem, ez az adott esetben nem megy.

5. A 3-4. képpel kapcsolatosak azonosak az 1. pontban leírtakkal, azzal kiegészítve, hogy a „vad" programok tették lehetővé a nagyvonalú memóriatérkép meghatározását.

(Én is zárójelben írom meg azt, hogy igen egyszerűen megvalósítható az, hogy az NMI — vagy bármilyen más — vezeték szoftver úton jelet kapjon. Mindössze annyi kell, hogy egy adott cím az adott vezetékre engedje például az órajelet.)

Végül nagyon örülök annak, hogy az olvasó ilyen sok és jól használható adatot közöl, amelyeket haladéktalanul közkinccsé teszünk.

BULGÁRIA

Klubok és iskolák

Bulgáriáról általában tengerparti üdülők, fejlett növénytermesztés, esetleg műemlékek, népművészet és a mienknél jóval fejletlenebb ipar jut eszünkbe. Ilyen ismeretek birtokában ez az írás meglepő vagy hihetetlen lesz.

A klubok

Azt tudtam, hogy Bulgáriában 3 éve központilag kezdtek számítógépes klubokat szervezni, és ezért az NTSZ (az MTESZ ottani megfelelője) több szakembert is küldött hozzánk, a mi klubmozgalmunk tanulmányozására. Tudtam azt is, hogy egy Apple II kompatíbilis gépet gyártanak iskolai terjesztésre. Ezek ellenére váratlan volt, amit hallottam, láttam. A tavalyi tavaszi BNV-n egy robotvezérlővel nálunk is bemutatták a bolgár gépet, amelynek fő előnye az Apple-kompatibilitás, mivel ehhez az amerikai géphez készült eddig a legtöbb mikroszámítógépes program a világon. Erről a gépről és a benne megvalósított technikai trükkökről külön cikkben számolunk be.

Bulgáriában két klubhálózatot hoztak létre: az egyiket a már említett NTSZ, a másikat a KISZ ottani megfelelője. Az első hálózatban 50 klub működik, ebből 12 Szófiában. A másodikról pontos adatot nem kaptam, csak annyit, hogy ennél több. A klubok helyiséget, berendezést, fizetett személyzetet kapnak. (Nálunk a HCC eddig még egyiket sem!) Öt klubban jártam, mindegyik hasonlóan volt berendezve, és egy gépteremből, társalgóból (három klub külön oktatóteremből is) állt. A géptermekhez 10-10 gépet kaptak, amelyek nagy része magnetofonnal volt ellátva, mindegyikhez zöld monitor tartozott, 2-3, egyenként 2-2,5" méretű lemezhez használható, hazai gyártású lemezegységgel rendelkezett, és volt 1-1 ugyancsak bolgár mátrixnyomtatójuk. Mint elmondták, az alapszoftvert, az oktatócsomagot is központilag terjesztik.

A klubok tevékenységének jelenleg meghatározó része az ismeretterjesztő tanfolyamok rendezése. Ezek kétfélék: az egyik díjtalan, 2-3 órás, a másik 60 órás, amelyért kb. 1000 forintnak megfelelő összeget kell fizetni, és amelynek keretében a géprendszerről általános ismereteket kapnak, megtanulják a billentyűzet kezelését. A BASIC nyelvet olyan szinten kell elsajátítaniuk, mint nálunk a TV-BASIC tanfolyam vizsgázóinak. Mindkét tanfolyamnál 2 emberre jut egy számítógép. A tanfolyamokon bárki részt vehet.

Egyelőre az önálló programírás melléktevékenység, és klubközi programcserét nem szerveztek. Ezeket későbbi feladatnak tartják. Gépépítés nincs a klubokban, és nem is tervezik. A gépek karbantartását, szervizét központilag végzik.

Állítólag lehet magántulajdonban külföldről hozott gép, de géptulajdonost senki sem ismert.

A nyári szünet miatt a meglátogatott öt klub közül csak kettő működött. Mindkettő KISZ-klub volt. Itt két csoporttal találkoztam: délelőtt és kora délután egy-egy fizető tanfolyam résztvevőivel, később pedig gyerekekkel, akik felügyelet mellett szabadfoglalkozást tartottak.

A tanfolyamokon a PRINT utasítást gyakorolták. Mivel a klub gépei elsősorban iskolai terjesztésre készültek, átváltható angol/cirill betűkészlettel vannak ellátva. Megtudtam, hogy a saját betűkészlet használatára kezdettől fogva megvolt a lehetőség, és vendéglátóim érthetetlennek tartották, hogy nálunk ez nem így van. Nem értették, hogy egy ilyen egyszerű és feltétlenül szükséges módosítást miért nem valósítottak meg. (Itt jegyzem meg, hogy a legújabb HT—1080Z változatnál — de a korábbiaknál továbbra sem — már valóban a teljes magyar betűkészlet megtalálható, és a betűk elhelyezése is csaknem megfelelő — csupán a hosszú alsó és felső szárú betűk érnek össze, ha egymás alá kerülnek. Sajnos a teljes magyar abc csak ezen a gépen létezik, és például az általános iskolai elterjesztésre szánt C16-on nem.)

Az egyik klubban a 14—17 éves gyerekek BASIC-programokat írtak gyakorlásként. A vezetőhöz csak akkor fordultak, ha problémáikon társaik sem tudtak segíteni. A programok néhány alaputasításból álltak, de volt olyan gyerek is, aki dallambeadásra alkalmas programot írt.

A másik klubban a tanfolyam hasonló volt, de a gyerekcsoport itt 10-12 évesekből állt, akiknek nagy része játszott, három gyerek pedig egyszerűbb feladatokat oldott meg a LOGO-nyelv teknősbéka-grafikájával. A klub vezetője elmondta, hogy inditanak egy kísérleti tanfolyamot, amelyen első nyelvként a PASCAL-t fogják oktatni. Céljuk ezzel, hogy a strukturált programok írásának elsajátításával programozóvá képezzék a hallgatókat.

Az iskolák

Szófiában tájékoztattak arról, hogy jelenleg 2500 Apple-kompatíbilis gép van az iskolákban, és jövőre 6000 lesz. A tervek szerint az általános iskolák is kapnak.

Egyetlen középiskolában jártam: nem fővárosi elit iskolában, hanem egy burgaszi átlagos, tagozatos gimnáziumban. A századelőn épített iskolában két számítógéptermet rendeztek be, az egyiket az oktatás, a másikat a gyakorlás céljára. Az oktatóteremben 18 gép van. Itt is láttam 4 lemezegységet és egy nyomtatót. Az oktatás során minden gyereknek külön gépe van. A gépek a tanári géppel és egy központi nagygéppel összeköthetők. A gyakorlóteremben további 9 gépet használhatnak a tanárok és a kiválasztott gyerekek.

Három tanár tanít 1200 gyereket. A matematika tagozatos gyerekek a különböző tantárgyakban évi mintegy 150 órát foglalkoznak valamilyen számítógéppel kapcsolatos dologgal. Számítástechnikában a BASIC-en kívül valamilyen közelebbről meg nem határozott FORTRAN-változatot is tanulnak. Gépi kódot szervezetten nem.

Bemutatták a legtehetségesebb diákjukat. 16 éves, két évvel ezelőtt az iskolában kezdett számítástechnikával foglalkozni. Jelenleg beszédszintetizáló programokat ír, egy BUSOFT nevű vállalkozás tagjaként. Az iskolában jártamkor a terménybetakarítás miatt szünet volt, így órákon nem vehettem részt.

—i —e

Ki ad magyarázatot?

A D100 típusú lengyel gyártmányú nyomtató igazán kellemes tulajdonságokkal rendelkező szerkezet. A HT iskolaszámítógéphez történő illesztése azonban talányos. Néhány egyszerű kérdés: Miért rontunk el egy kényelmesen működő nyomtatót egy olyan kényelmetlen illesztővel, amely külön illesztőprogramot igényel? Egyetlen kód konvertálása olyan nagymértékben megdrágította volna, hogy az iskolák nem tudnák megfizetni?

Természetesen ez „csak" kényelmetlen megoldás, de nem használhatatlan. Úgy értem: a megoldás használható, de a program nem. Miért kerül a gép mindig hidegstartra, ha stringművelet (például összeadás) után nyomtatási parancsot adunk? Lehet, hogy az illesztőprogram nem illeszkedik?

TRÉNING-MÓDOK

A játékprogram-rajongók gyakori problémája, hogy az innen-onnan beszerzett játékokat kezdetben csak igen körülményesen tudják használni. A klubokban, cseretársaktól másolt programokhoz ugyanis nincs leírás, ezért a játékos csupán találgatja a feladatot.

Előfordul, hogy a játék túl nehéz és kevés az életszám, így élvezet helyett kínlódást jelent a próbálkozás. Ezen kívánunk segíteni az alábbi felsorolással, amely néhány népszerű játék „örökéletűvé" tételét mutatja a két legelterjedtebb mikroszámítógépre.

ZX—SPECTRUM

Booty	10 BORDER 0:PAPER 0 :INK 0:CLEAR 26870 20 LOAD ""SCREEN$:BORDER 0:PAPER 0 :INK 0 30 PRINT AT 19,0 :LOAD ""CODE 26880 40 RANDOMIZE USR 26880 :POKE 58294,0 50 RANDOMIZE USR 52500
COOKIE	POKE 28698,0
Eskimo Eddie	POKE 24686,24 POKE 24687,76
Fantastic Voyage	10 CLEAR 30719 20 LOAD"VOYAGE" CODE 30 POKE 54492,0 :POKE 54227,0 40 BORDER 0 :PRINT USR 53248
Hunchback	POKE 26888,0
Jetpac	POKE 25020,0
Jet Set Willy	35 POKE 35899,0(ENTER), MERGE ""(ENTER). Betöltés után RUN, szalag indul
Lunar Jetman	POKE 36966,224 POKE 36945,3
Manic Miner	A loaderben az indítás elé: POKE 34269,32 (max!)
Tranz Am	POKE 25446,0
Sabre Wolf	A BASIC loaderben az első sor editálásához direkt parancsként: POKE 23756,1:CLEAR 65535. A sor lehozása után töröljük a sorból a PRINT USR 23424-et. A 10-es sorba: POKE 43575,255:POKE 45520, 255:POKE 45599,255 :POKE 41725,255. A 20-as sorba: PRINT USR 23424

COMMODORE 64

Néhány programnak több változata forog közkézen, így nem garantálhatunk minden esetben biztos sikert. Ha több változatról is tudomásunk van, a vagy szócskával utalunk rá. A POKE-ok — csak a címeket és a tartalmat jelezzük — betöltés után, indítás előtt esedékesek.

Annihilator	6295,11
Blagger	3560,8
Burnin Rubber	18432,173
Bungling Bay	47465,176
Buck Rogers	8825,36
Black Hawk	8289,99
Bat Attack	11061,234
Battlezone	8909,100
Bruce Lee	5686,128 és 5677,128
China Miner	34623,234 34624,234 vagy 33301,245 és 33457,255 vagy 34634,44
Choplifter	8011,173
Crazy Kong	30624,173
Crossfire	27625,173 vagy 5353,44
Chrisis Mountain	2665,238 vagy 3144,238
Clowns	3566,255
Dare Devil Dennis (tempóállítás:	29173,255 17958,T alapérték = 5)
Dimension X	8645,129
Dinkey Doo	11989,99 vagy 11989,18
Dig Dug	10473,255
Donkey Kong	11989,99 vagy 11989,18
Falcon Patrol	A főprogram előtt: 16764,36:16705,2:SYS 5640 Név: FALCON PATROL vagy 16764,234 és 16765,234
Fort Apocalypse	36339,153 vagy 14697,0:14760,0:36,366,0
Frantic Freddie	34535,24
Fire Ant	17568,100
Frogger	22341,173
Frogger Sega	22341,173
Galaga	17388,173
Galaxions	7065,230 vagy 17288,165
Gangster	5989,58
Galaxy	3369,230 vagy 3378,230
Hard Hat Mack	16877,173
Hunchback	9521,44 vagy 9521,234:9522,234 :9523,234:
Herby	7191,255
Jumping Jack	27904,173
Jumpman Junior	9540,44
Jungle Hunt	2242,234:2243,234
Jet Set Willy	11345,33
Kaktus	4564,255
Kid Grid	10020,234
Laser Strike	16475,173
Lady Tut	2392,50
Lazy Jones	2971,9
Manic Miner	A főprogram előtt: 16571,173:SYS 16384 Név: MM
Moon Buggy	24151,173
Miner 2049er	9450,173
Matrix	7629,238 vagy 7983,238
Motor Mania	8646,255
Neptune	7870,60
Pogo Joe	2779,36
Pooyan	20634,173
Pengo (Petch)	20295,44
Pedestrian	2288,255
Pitfall (Activision)	5393,255
Pakakuda	7015,234
Q-Bert and R-Nest	4446,173
Quest for Tires	7341,99 vagy 11485,125
Radar Rat Race	7194,234
Revenge Camels	2599,230 vagy 2746,230 vagy 39931,238
Roundabout	12843,234
Revenge of the Mutant Camels	35518,250
Robin to the Rescue	6144,234
Sammy Lightfood	3678,189
Seafox	7337,173
Shamus	18486,169 vagy 23558,169
Shamus Case 2	15476,176 vagy 15475,238 vagy 3888,128
Snokie	33242,255 vagy 33242,200
Space Taxi	RUN/STOP—RESTORE után 16911,200
Sheep in Space	35039,44
Squish,em	2562,100
Zeppelin	18546,44 vagy 10081,100 vagy 14337,100
Ghostbusters	Betöltés után név: Martin, Account: yes Szám: 02203505. Eredmény = 150 200 dollár. vagy Név: MIX, Account: yes Szám: 60747400. Eredmény = 33 000 dollár.

PINTÉR TIBOR

UTCÁN ÁT...

METRO-COMP
Budapest, VI.,
Bajcsy-Zsilinszky út 63.
Telefon: 127-442

Ezt az üzletet már bizonyára sokan ismerik, hiszen 1984. április 4-én, a Skála Metróval egy időben nyitották meg. Profilja tisztán bizományosi. Forgalmának kb. 80 százalékát adják a számítógépek és alkatrészek.

Árképzésük hasonló a BÁV-éhoz. Előnyük, hogy főleg készpénzért vásárolnak, így valamelyest lerövidül a termék forgalmi átfutási ideje.

Érdekes és hasznos tevékenységük az ún. közületi vevőszolgálat: levélben értesítik a vállalatokat, felhasználókat a boltban kapható gépekről, alkatrészekről. Ugyanakkor igyekeznek információt szerezni a piaci keresletről és a magánbehozatalt annak megfelelően orientálni.

Jelenleg tárgyalnak a boltban kapható gépek szervizproblémáinak megoldásáról egy kisszövetkezettel. Ennek értelmében fél év garanciát biztosítanának az itt vásárolt C64 berendezésekre, hiszen forgalmuk döntő részét ez a géptípus teszi ki.

Egy másik kisszövetkezettel a C64 gépekhez 10-15 Mbájtos Winchester illesztéséről tárgyalnak.

Csak náluk kapható a 1541-es floppymeghajtóhoz a FLOTISZ, többször felhasználható tisztítólemez.

Primóra

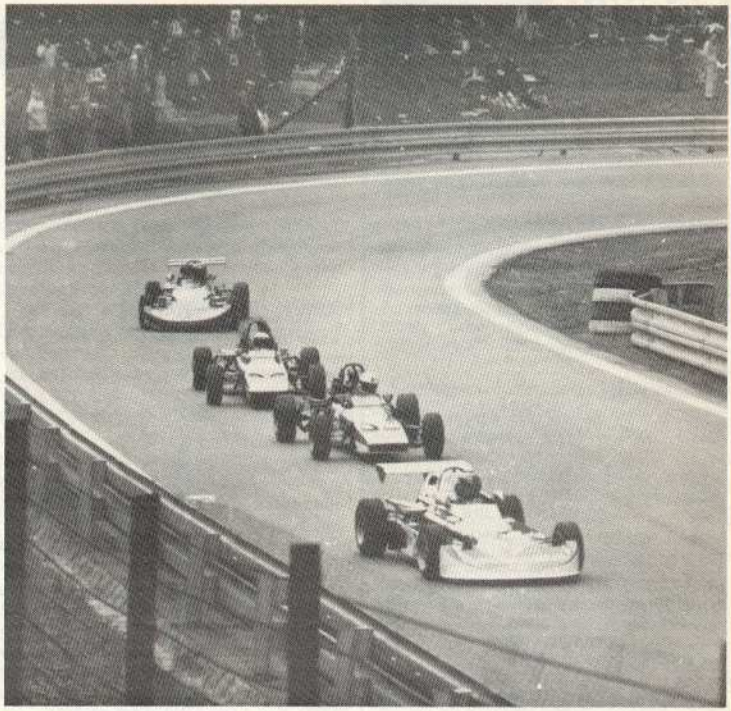
FORMA I.

Egy kanyargós pályán minél nagyobb távot kell megtennünk, megelőzve a többi versenyautót. Saját járművünket az ú és az Y billentyűkkel irányíthatjuk; ezek lenyomásáig autónk egy kanyarodó jármű képét mutatja. Ha akadálynak vagy a pálya szélének ütközünk, megtudjuk az addig megtett út hosszát. A pálya középre zárkózik, és kezdhetjük elölről a játékot.

A program felépítése:

5	a gép átállítása nagybetűs írásmódra
10-30	változók és speciális karakterek definiálása
35	játékszabályok
45-55	a kanyarodás ívének meghatározása, a kikerülendő autó megjelenítése
65	hanghatás, saját autónk kirajzolása, scroll
75-80	irányítás
85	ütközés figyelése
95-115	ütközés utáni hanghatás, az eredmények kijelzése, a pálya középre zárkózása

MARKÓ GYÖRGY



```
5 CLS: IF PEEK(16452)<128 THEN POKE 16452,PEEK(16452)+128
10 A%=18: B%=0: C%=0: I%=1: L%=1: R%=0: U%=15: V%=0: DIM A$(2): U$="#
#"
15 C=0: DIM K%(16): C=VARPTR(K%(0))
20 POKEC,0,0,0,6,0,12,14,13,0,0,24,102,36,126,126,25,0,0,0,96,0,48,112,49,62,2,1
18,126,119,0,0,0,112,32,92,114,126,108,30,25,0,0,0,0,64,104,60,109,0,0,0,0,2,22,
60,55,14,4,58,78,126,54,120,25,124,64,110,126,111
25 POKE 16459,C-256*INT(C/256),INT(C/256)
30 A$(2)=CHR$(131)+CHR$(132)+CHR$(133): A$(1)=CHR$(128)+CHR$(129)+CHR$(130): A$(
0)=CHR$(134)+CHR$(135)+CHR$(136)
35 PRINT$ 0,0,A$(2)TAB(13)CHR$(2)"Forma 1"CHR$(18)TAB(39)A$(0): PRINT$ 4,5,"Irán
yítás: balra = Y  jobbra = ú": FOR B%=0 TO 4000: NEXT B%
40 CLS: GOTO 60
45 V%=RND(3)-2: PRINT$ 14,U%+6-(V%+1)*2,A$(1)
50 IF U%<1 THEN V%=1
55 IF U%>24 THEN V%=-1
60 FOR B%=1 TO 5
65 BEEP 150,10: PRINT$ 14,U%,U$: PRINT$ 15,0,"": PRINT$ 0,A%,A$(I%): IF C%<0 THE
N B%=6: GOTO 90
70 I%=1: U%=U%+V%: L%=L%+1
75 A%=A%-(INKEY$="ú")*2+(INKEY$="Y")*2
80 I%=I%-(INKEY$="ú")+(INKEY$="Y")
85 C%=POINT(6*A%+8,175)+POINT(6*A%+12,175)+POINT(6*A%+17,175)
90 NEXT B%: IF B%<7 THEN 45
95 FOR B%=40 TO 100 STEP 4: BEEP B%,B%/4: NEXT B%
100 FOR B%=1 TO 20: U%=U%-(U%<15)+(U%>15): PRINT$ 14,U%,U$: PRINT$ 15,0,"": PRIN
T$ 0,37,L%
105 IF L%>R% THEN PRINT$ 0,0,"UJ REKORD!"
110 NEXT B%: IF L%>R% THEN R%=L%
115 A%=18: C%=0: I%=1: L%=1: GOTO 45
```

```
EA60  210000    LD    HL,0000
EA63  E5        PUSH  HL
EA64  110300    LD    DE,0003
EA67  CDB503    CALL  03B5
EA6A  E1        POP   HL
EA6B  23        INC   HL
EA6C  10F5      DJNZ  EA63
EA6E  C9        RET
EA6F  00        NOP
EA70  00        NOP
EA71  00        NOP
EA72  00        NOP
EA73  00        NOP
EA74  00        NOP
EA75  00        NOP
EA76  00        NOP
EA77  00        NOP
EA78  00        NOP
EA79  00        NOP
EA7A  00        NOP
EA7B  00        NOP
EA7C  00        NOP
EA7D  00        NOP
EA7E  00        NOP
```

```
EE48  F3        DI
EE49  3E07      LD    A,07
EE4B  06FF      LD    B,FF
EE4D  C5        PUSH  BC
EE4E  C5        PUSH  BC
EE4F  EE10      XOR   10
EE51  D3FE      OUT   (FE),A
EE53  10FE      DJNZ  EE53
EE55  C1        POP   BC
EE56  EE10      XOR   10
EE58  D3FE      OUT   (FE),A
EE5A  10FE      DJNZ  EE5A
EE5C  C1        POP   BC
EE5D  10ED      DJNZ  EE4C
EE5F  FB        EI
EE60  C9        RET
EE61  00        NOP
EE62  00        NOP
EE63  00        NOP
EE64  00        NOP
EE65  00        NOP
EE66  00        NOP
```

ZX—Spectrumra

HANGHATÁSVADÁSZOK FIGYELMÉBE

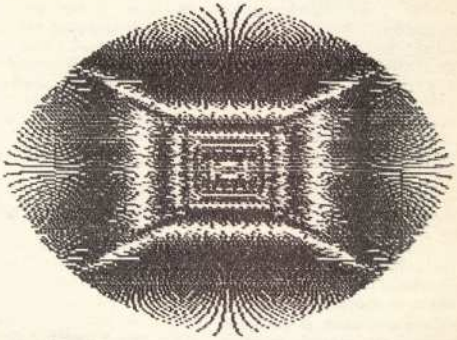
A fenti decimális listát egyszerűen be lehet építeni a BASIC-be. A könnyebb megértés kedvéért közöljük az assembler listát is.

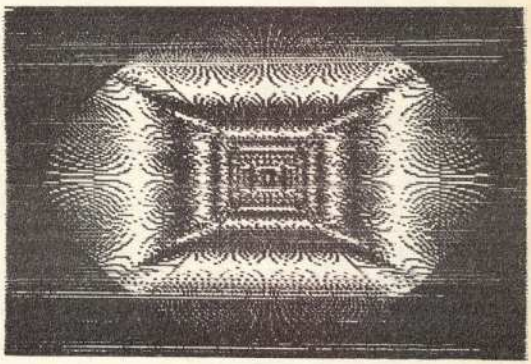
1. 33Ø,Ø,229,17,3,Ø,2Ø5,181,3,225,35,16,245,2Ø1
2. 243,62,7,6,255,197,238,16,211,254,16,254,193,238,16,211,254, 16,254,193,16,237,251,2Ø1

BAKSA ZSOLT

ZX—Spectrumra

PROGRAMGRAFIKA





```
  5 BORDER 0: CLS : LET x=1
 10 OVER 1: LET x=x-3*(x>3)
 20 FOR n=0 TO 175 STEP x
 30 PLOT 0,n: DRAW 255,175-2*n
 40 NEXT n: FOR n=0 TO 255 STEP
x
 50 PLOT n,175: DRAW 255-2*n,-1
75
 60 NEXT n: OVER 0
 70 FOR n=0 TO 175 STEP x
 80 PLOT 0,n: DRAW n,175-n
 90 PLOT 0,175-n: DRAW n,n-175
100 PLOT 255,n: DRAW -n,175-n
110 PLOT 255,175-n: DRAW -n,n-1
75
120 NEXT n: LET x=x+1: GO TO 10
```

Beküldte: BAKSA ZSOLT

SZÁMÍTÓGÉP-FEJLESZTŐK! GYÁRTÓK!

Nálunk azonnal kapható MOM 1800/900 típusú 5¼" hajlékonylemezes meghajtóegység. Tápegység és doboz nélkül ára 18 000,— Ft.

Érdeklődni lehet a 252-880-as telefonszámon

Közlekedő- és

Számolási feladatokat tömegesen ma már kizárólag zsebszámológépekkel, személyi és nagyobb teljesítőképességű elektronikus digitális gépekkel oldanak meg. Ez azzal jár, hogy majdnem teljesen kiszorult a köztudatból az, hogy vannak nem elektronikus és nem digitális számolóeszközök is. Pedig sok érdekes, szellemes, sőt olyan darab is van közöttük, amely saját feladatkörében még ma is a leghasználhatóbb, és jó ideig még az is marad.

A tanulságos és szellemes, de elavult, vagy soha professzionálisan nem használt eszközökre a szórakoztató elmetornán kívül azért is érdemes néhány órát szánni, mert a bennük levő ötletek — a matematikai modellezési készségünk fejlesztése révén — sokat segíthetnek, különösen a nem szokványos, nehezebb műszaki problémák megoldásában.

A máig is használt, de újabban már csak kevés számítástechnikus által ismert, és még kevesebbnek a mindennapi munkájában szerepet kapó, nem elektronikus digitális eszközökkel és gépekkel való megismerkedés még azért is szükséges, hogy mindig pontosan tudjuk a digitális elektronika alkalmazásának pillanatnyi határait, azt, hogy melyek azok a területek, ahol ezek a más jellegű segédeszközök hatékonyabbak az egyebütt általánosan használt digitális elektronikusaknál.

A számolóábrák (nomogramok) és hibrid számológépek példája bizonyítja, hogy vannak nem (teljesen) digitális elvvel dolgozó eszközök, amelyek még ma is szilárdan tartják helyüket a számolási munkát megkönnyítő műszaki és szellemi alkotások között. Célszerűbb azonban, ha először nem ezekkel a „nehézsúlyú" professzionális konstrukciókkal, hanem néhány egyszerű és szellemes fizikai számolómodellel ismerkedünk meg. Ebből azonban ne vonjunk le olyan elhamarkodott következtetéseket, hogy a számítástechnikának kizárólag a fizika az eszközbázisa. Hiszen éppen napjainkban teszi első lépéseit a biológiai és kémiai elveket hasznosító automaták elmélete és gyakorlata is. A fluid automaták chemoton elméletéről például nemrégiben már könyv is jelent meg magyar nyelven.

Azt eddig is tudtuk, hogy ma még műszakilag utánozhatatlanul nagy teljesítőképességű kémiai és biológiai automaták működnek az élővilágban, most azonban már ilyenek mesterséges létrehozására is megindultak az erőfeszítések. Nincs kizárva, hogy eszközeink között rövidesen megjelennek majd a nemcsak fizikát hasznosító konstrukciók is.

Vízszint lábasban, vályúban, tölcsérben

Öntsünk vizet egy lábasba vagy egy mérőedénybe! Ha az edények belsejét ideális hengernek tekinthetjük, akkor a folyadék szintmagassága arányos lesz a beléjük öntött folyadékmennyiséggel. Ha a szintmagasság x, az edénybe öntött folyadék térfogata vagy tömege pedig m, akkor

$m = c \cdot x$, illetve $x = \frac{m}{c}$,

ahol c az edény alapterülete. A rendszer vázlatát és kapcsolatábráját szemlélteti az *1. ábra*.

Öntsünk most m tömegű vizet egy háromszög keresztmetszetű vízszintes vályúba! A vályú adatait és a rendszer kapcsolatábráját a *2. ábra* szemlélteti. Mekkora lesz a szintmagasság a vályúban? Nyilván $m = tg\alpha \cdot l \cdot x^2$.

Ha az x-től független $tg\alpha \cdot l$-et b-vel jelöljük, akkor $m = b \cdot x^2$, és $x = \sqrt{\frac{m}{b}}$.

A vályúba öntött folyadék szintmagassága tehát arányos lesz a folyadék tömegének négyzetgyökével. Ha a vályút úgy készítjük el, hogy b értéke 1 legyen, akkor $m = x^2$, és $x = \sqrt{m}$.

Megállapíthatjuk tehát, hogy szert tettünk egy hidrosztatikai négyzetre emelő és gyökvonó gépre. Ha ugyanis m négyzetgyökére vagyunk kíváncsiak, akkor m tömegű vizet öntünk a vályúba, és leolvassuk a szintmagasságot. Ha pedig x négyzetére van szükségünk, akkor addig öntünk vizet a vályúba, amíg az x magasságú nem lesz. A vályúban levő víz mennyisége ilyenkor x^2 értékével fog megegyezni.

Ugyanígy végezhetjük el harmadik hatványnak, illetve köbgyöknek a kiszámítását (helyesebben mérését), ha a vályú helyett gúlát vagy kúpot, például tölcsért használunk.

Kijelentésünket egy egyenes, négyzet alapú gúla *(3. ábra)* esetében a következőképpen bizonyíthatjuk be. A gúla alaplapjának területe $(2 \cdot x \cdot tg\alpha)^2$. Magassága x. Térfogatára (tömegére) tehát fennáll, hogy

$m = \frac{(2 \cdot x \cdot tg\alpha)^2 \cdot x}{3}$,

amiből

$m = \frac{4tg^2\alpha}{3} x^3$.

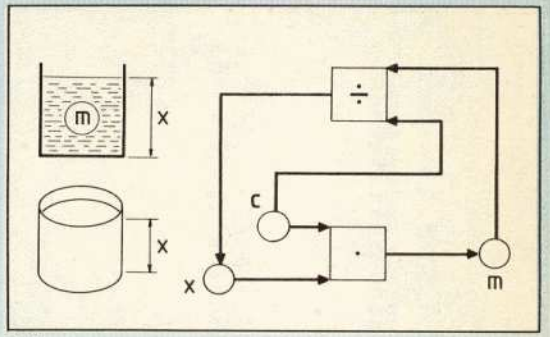
Ha az x-től független $\frac{4tg^2\alpha}{3}$ tényezőt a-val jelöljük, akkor

$m = ax^3$, és $x = \sqrt[3]{\frac{m}{a}}$.

Ha pedig olyan gúlát szerkesztünk, amelyre a értéke 1, akkor $m = x^3$, és $x = \sqrt[3]{m}$.

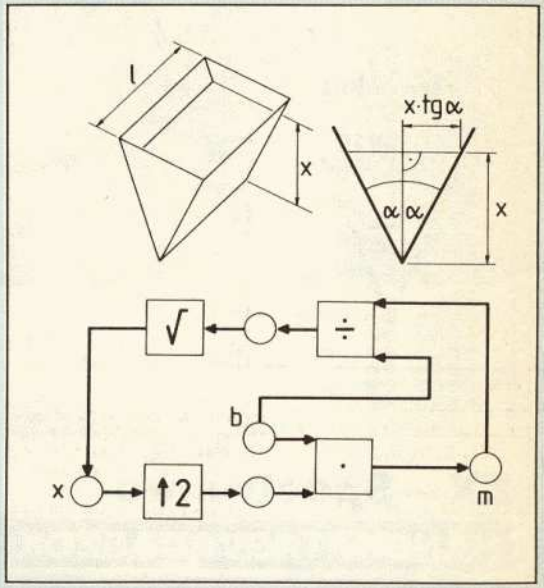
A *4. ábra* néhány közlekedőedényt és a velük megoldható egyenleteket szemléltet. Az *5. ábra* közlekedőedényének egyenletében pedig minden fokszámú tag előfordul. A közlekedőedények axonometrikus vázlatán az alsó összekötő vezetéket egyetlen vonallal szemléltettük. A félreértések elkerülése végett a 4. ábra középső közlekedőedényének elvi metszetét is megrajzoltuk *(6. ábra)*.

Az egyenletekben x a folyadék szintmagassága, m pedig az össz-folyadékmennyiség, amit a közlekedőedénybe ön-

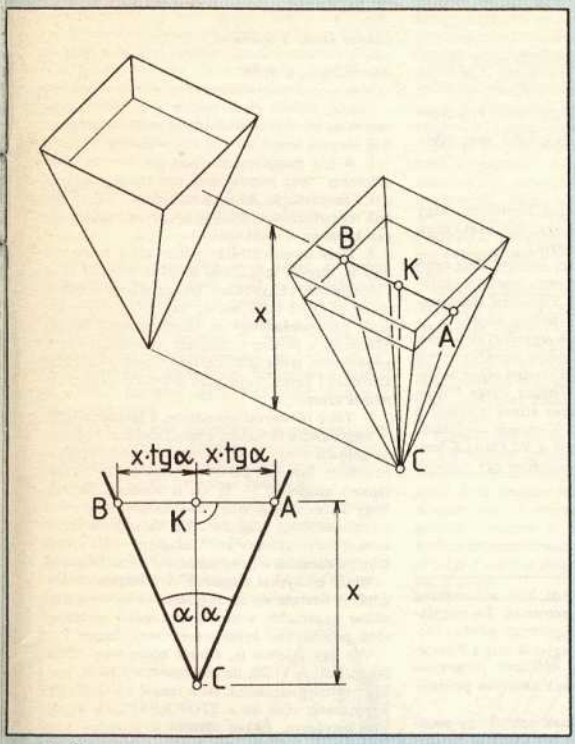


1. ábra

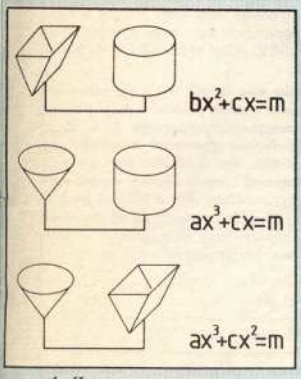
2. ábra



számolóedények

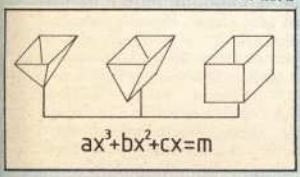


3. ábra



4. ábra

5. ábra



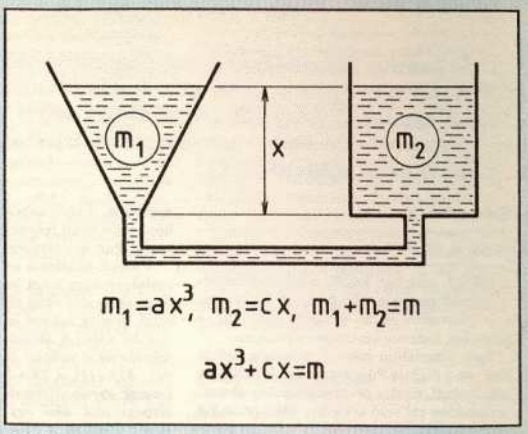
töttünk. Az összekötő vezeték térfogatát elhanyagolhatónak vehetjük, vagy előre feltöltött vezetékkel kezdjük a mérést.

Az egyenletek azt fejezik ki, hogy az egyes edényekben levő folyadék összesen *m* tömegű. A tömeget — mivel vízzel dolgoztunk —, azonosnak vettük a térfogattal.

Az 5. ábra közlekedőedényébe *m* tömegű vizet öntve, az $ax^3+bx^2+cx=m$ egyenlet egy pozitív megoldását olvashatjuk le.

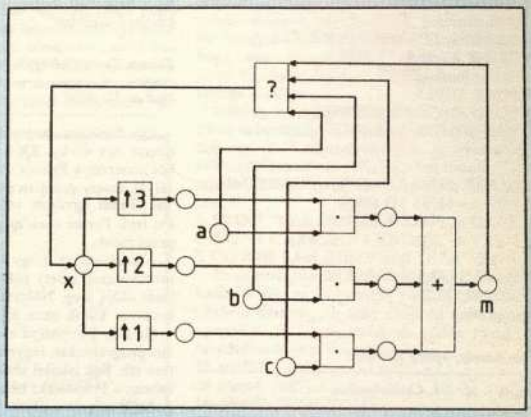
A rendszer *7. ábrán* látható kapcsolatábráján csak az *x*-ből *m* felé irányuló számolás rendjét tudtuk részletesen ábrázolni. Az inverz feladat, amit a kérdőjellel jelölt operátor végez, a harmadfokú egyenlet gyöke(i) kiszámítási rendjének részletes ábrázolása, noha lehetséges, meglehetősen bonyolult. Ez is indokolja a más úton való gyökkeresési kísérleteket.

Ahogy a példák is bizonyítják, a bemutatott közlekedőedényeket — a számolóábrák mintájára — teljes joggal számolóedényeknek is nevezhetjük.



6. ábra

7. ábra



Hogy hol a határa a számolóedényekkel való szellemes manipulációknak, nem tudjuk. Azt azonban nyugodtan állíthatjuk, hogy világuk gazdag, és még sok érdekes modellt rejteget.

A 6. ábra modellje a számítástechnikai irodalomban ismereteink szerint legalább 20 éve fel-felbukkan itt-ott. Pontos eredetét nem ismerjük. Az ilyen modellek rendszeres összefoglalásával még nem találkoztunk.

További érdekes kérdések

Milyen számolóedények készíthetők háromnál magasabb fokú polinomok gyökeinek mérés útján való megállapítására?

Eddig csak nemnegatív együtthatókat szerepeltettünk, az állandó pedig sosem volt pozitív. Mi a helyzet a nem vizsgált esetekben?

Eddig csak polinomokkal foglalkoztunk. Más függvényekkel kapcsolatos számolóedények is szerkeszthetők?

Lehet-e olyan modellt szerkeszteni, amely a megadott *x*-hez tartozó helyettesítési értéket szolgáltatja? (Helyettesítésiérték-számoló edény.)

Ókori nagyok felismerése volt, hogy az istenek aritmetizálnak. A középkorban észrevették, hogy gyógyítani a természet gyógyít. Ma — példáink alapján — kimondhatjuk, hogy a természet számol (is).

N. A. P.

KEDVES OLVASÓINK!

Lapunk évente tizenkétszeri megjelenésével lehetőség nyílik több cikk, program, ötlet közlésére. Magazinunk színvonalának emelése érdekében az Önök aktív közreműködésére is számítunk. Ezért várjuk, hogy írjanak. Sokat és jól. És most lássuk a leveleket!

Dr. Szász Gábor, Budapest,

Juhász Gy. u. 10. 1039

Az 1985/3. számban közölt repülésszimulációs programok percekig töltődnek, így ha van valakinek microdrive-ja, és gyakran használja a programokat, érdemes cartridge-ra menteni.

A Flight Simulation minden nehézség nélkül kitehető, de a Fighter Pilot esetén egy kis trükköt kell alkalmazni, mert a programtöltéskor elrontja a microdrive-ról való olvasást. Először az FP betöltőprogramot kell cartridge-ra tenni (1. program), majd a 2. programot kell beírni a gépbe. Ez beolvassa a kazettáról a főprogramot, és cartridge-ra menti.

```
1Ø BORDER 1:PAPER 1:INK 7:CLS
2Ø PRINT AT 1Ø,9; FLASH 1;
   "FP is loading"
3Ø POKE 23693,9
4Ø LOAD •"m";1;"fpm"CODE
5Ø RANDOMIZE USR 63793
```

1. program

```
1Ø CLEAR 29999: LOAD "fpm"CODE 3e4
2Ø FOR n=63793 TO 638Ø4
3Ø READ a: POKE n,a
4Ø NEXT n
5Ø SAVE•"m";1;"fpm"CODE 3e4,3 38Ø5
6Ø DATA 33,48,117,17,Ø,92,1,Ø,
   132,237,176,2Ø1
```

2. program

Csúz Károly, 98401 Lučenec,

Május 1. tér III. Csehszlovákia

Hű olvasója vagyok lapjuknak, de mivel külföldön élek, sajnos néha nagyon nehezen tudok hozzájutni. Mivel nálunk nincs hasonló jellegű folyóirat, nagy boldogsággal töltött el, mikor egy magyarországi kiránduláson véletlenül meglátam egy újságárusnál. Ha néha-néha hozzájutok egy számhoz, legalább 20 barátom vár rá, hogy áttanulmányozhassa.

A szerkesztőségi cikkben is megírtam, hogy a következő nagy feladatunk az újság külföldi terjesztésének megszervezése. Remélem, akkor mind a húsz barátja megkapja a lapot. A kért és hiányzó példányok közül az egyiket elküldjük, a 83-as számhoz már nem lehet hozzájutni.

Kovács Zsolt, Hajdúszoboszló,

Luther u. 20/B 4200

Sikeresen letettem a TV—BASIC-vizsgát, és szeretném megtudni, hogy ez mire jogosít fel engem. Ebben az évben fogok érettségizni, így a pályaválasztás gondja egyre súlyosabban nehezedik rám. A bizonyítvány szövegében ez áll: „. . . valamennyi SZÁMALK tanfolyamon BASIC programozási nyelvi vizsgának, illetve tanfolyami felvételi vizsgának ismerik el . . .” a bizonyítványt. Ezzel kapcsolatban kérdezem, hogy hová lehet ezzel felvételizni, azaz hol lehet ennek birtokában tanfolyamot végezni?

A másik kérdésem az, hogy mennyi ideig tartó tanfolyamokra lehet jelentkezni, illetve a tanfolyam elvégzése után milyen képesítést lehet szerezni, esetleg miként lehet elhelyezkedni?

A SZÁMALK tanfolyamok jegyzékét elküldtük, jelentkezni a vállalat tanulmányi osztályán lehet (tel.: 853-111). A TV—BASIC-vizsga olyan tanfolyamok esetén mentesít a felvételi vizsga kötelezettsége alól, ahol egy magas szintű nyelvből a vizsga kötelező. A TV—BASIC-vizsgát elfogadják magas szintű nyelvi vizsgának a SZÁMALK-ban, ha a választott tanfolyam keretében egy nyelvből kötelező a vizsga.

Takács Zsolt, Nyíregyháza,

Nyíl u. 24. 4400

Egy barátom megbízásából írok, aki nemrég hozott egy 48 k-s ZX—Spectrumot. Én nagyjából ismerem a Primót, ő a Spectrum néhány fortélyát, amely azonban nem egyezik meg a Primóéval. Tehát együttes erővel szoktunk programokat írni. Persze ezek még csak amolyan primitív programok.

A kérésem az, hogy írjanak nekem egy programrészletet, amely például egy karakter mozgatását oldja meg. Nekünk még egy pontot is csak keserves kínok után sikerült mozgásra nógatni.

Ha már a Primóról esett szó: hiányolom a Primo-programokat, legyen az játékprogram, bemutató stb. Sok iskolai szakkör előlegezte meg a bizalmat a Primónak; biztosan ezek is szívesen fogadnák az erre a gépre írt programokat.

Kérését teljesítjük: keresünk valakit, aki a kívánt programot megírja, és a programrészletet elküldjük. Ami a Primo-programokat illeti, szívesen közölnénk bármilyen programot, ha küldenek; mi abból dolgozunk, amit elsősorban olvasóinktól kapunk. Primóra — ki tudja miért — nagyon kevés program érkezik. Talán ez a kis írás megmozgatja a Primo-tulajdonosokat, és elárasztanak bennünket programokkal.

Bíró Lajos, Revolučna 972/A5 92400

Galánta, Csehszlovákia

Már a megjelenésétől kezdve nagy érdeklődéssel olvasom a Mikroszámítógép Magazint. Sajnos nincs meg az összes száma, mert Csehszlovákiában nem kapható, és a beszerzése nehézségekbe ütközik.

Én egy SORD személyi számítógép tulajdonosa vagyok, és ahogy tapasztaltam, Magyarországon ez a márka kevésbé ismert, ezért gondoltam, hogy bemutatom.

Ezután egy igen jó leírást kaptunk a SORD M5-ös számítógépről. Nem tudom, hogy közöljük-e vagy sem, ti. nem hallottam, hogy ilyen gép volna az országban. Így számomra kétséges, hogy a leírás érdekli-e olvasóinkat. Megnézzük, a cikket köszönjük.

Lukács Attila, Salgótarján,

Alkotmány u. 1. 3100

Lapjuk érdemi része mellett előszeretettel olvasom az olvasói rovatot is. Az ott felvetett gondok alapján lenne néhány észrevételem.

1. A lap megjelentetésében azt hiszem több „illetékes” még mindig nem érti ennek a dolognak a fontosságát. Az évi 6 megjelenés aligha látszik teljesíthetőnek, amikor a 4. szám október végén kerül az olvasókhoz!

2. Ilyen hosszú átfutás mellett talán lenne idő arra is, hogy a megjelenő hibák a nyomás előtt kiderüljenek. Egyébként tapasztaltam önöknél is, de az Ötlet-Bitletben is, hogy a printeren készült programlistákba is becsúszhatnak hibák. Márpedig a printer elvileg azt nyomtatja, ami a gépben van, tehát talán „direkt” rossz programot közölnek? Lehet, hogy nem a szerkesztőség, hanem a szerző.

3. Több olvasóval egyetértve, a javulás ellenére még mindig keveslem a programokat. Jó lenne megoldani olyan programok közlését is, amilyenekről dr. Simonyi Endre ír, hogy külföldi szaklapban megjelent — és ott is hibásan. Sejtem, hogy a szerzői jogi okok és a devizahiány miatt ez nem oldható meg, de talán valamiféle cserebere révén mód lenne rá. A plágium vádja nélkül lehetne ezeknek a programoknak az átdolgozott — de az erényeket megtartó — változatát közreadni. A devizás jogdíjmegszerzés szerintem még akkor is olcsóbb, mint ha az eredeti szaklapot több példányban kellene megvenni, importálni.

Van egy ötletem is, amiről eddig nem olvastam sehol! A VC20, de gondolom a C64 is, amikor bevitelre várakozik, nem reagál a STOP megnyomására, csak ha a STOP-RESTORE együtt van lenyomva. Ekkor viszont a program nem folytatható. Felfedeztem, hogy a gép két SHIFT billentyűje nem ugyanaz. A jobb oldali SHIFT és a STOP lenyomása meg tudja állítani a bevitelre váró programot is úgy, hogy nem hajtódik végre a RESTORE. A bal oldali SHIFT és STOP ezt nem tudja.

Hosszú levelét nem tudjuk teljes egészében közölni. A kihagyott részben például a magas alkatrész- és berendezésárakat kifogásolja. Mi is, de ez ma a realitás, keveset gyártanak mindenből, és ezért nagyon drága. Ami a hibákat illeti, szorgalmasan „bogarászunk”, ennek ellenére marad egy-egy hiba a programokban. Ami a külföldi programokat illeti, hamarosan erre is sor kerül, megtaláljuk a módját annak, hogy más lapoktól is átvegyünk írásokat és persze programokat is.

Honti József, Csákvár,

Május 1. út 11. 8083

Községünkben is megalakult egy mikroklub: Mikroklub és Számítógépbarátok Köre. A megnevezésben azt is kifejezésre akartuk juttatni, hogy nem zárt körű klubot, hanem minden érdeklődő számára nyitott kört kívántunk szervezni. A klub üzemeltetését a Kossuth Művelődési Ház vállalta magára. A klub vezetője Hujber Antal (Csákvár, Széchenyi út 1. Telefon: 06-24-44-236). A klub 5 Commodore C16-os géppel rendelkezik. Tagdíj nincs, a tanfolyamokért viszont fizetni kell. A tanfolyamok önköltségesek, tehát a tanfolyami díj a felmerülő költségek függvényében változó.

A programcsere megszervezésével engem biztak meg. Sajnos a C16-osokhoz semmiféle programunk nincs, és nincs gyári DEMO-kazettánk se. Rendelkezésünkre áll viszont C64-re írt 200 db játékprogram, közel ennyi felhasználói program. Ezeket szívesen bocsátjuk azok rendelkezésére, akik C16-os programokat tudnak cserébe adni.

Örömmel köszöntöm az új klubot, és szívesen közöljük felhívásukat.

Frigyes Ervin,

Tervgazdasági Kutató Intézet

Vissza az ékezetekhez című írásuk késztetett arra, hogy beszámoljak arról a kísérletemről, amelynek eredményeként például ezt a szöveget Sinclair ZX—Spectrum gépen, a TASMAN Szoftverház TASWORD 2 word processzor programjának kiegészítésével nyomtattam ki Centronics GLP nyomtatón. Ez utóbbi a külföldi piacon kapható egyik legolcsóbb kompakt mátrixnyomtató.

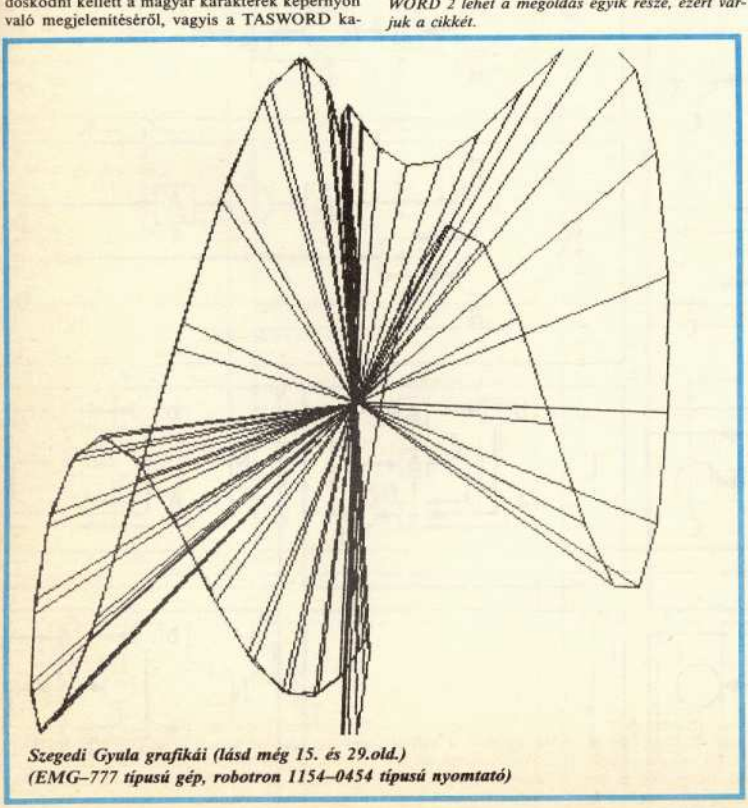
Mivel a sok más nyomtatóra is alkalmazható, kizárólag szoftvereszközökre támaszkodó eljárás esetleg másokat is érdekelhet, úgy érzem, helyes, ha lapjuk nyilvánosságához fordulok.

A szövegfeldolgozónak magyar karakterek használatára való alkalmazása — printer„átégetés" nélkül — kettős feladat volt. Először gondoskodni kellett a magyar karakterek képernyőn való megjelenítéséről, vagyis a TASWORD karakterkészletébe (64 karakter/sor) néhány Symbol shiftes ASCII karaktert feláldozva, el kellett helyezni az „ékes" betűket. Másodszor gondoskodni kellett olyan karakterkód-transzformáló nyomtatóprogramról, amely a feláldozott ASCII karakterek kódjait a printer megfelelő kódjaira leképezi, és elintézi ezek kinyomtatását a TASWORD 2 kiváló szerkesztési és nyomtatásvezérlő lehetőségeinek hasznosításával.

Mivel maradt még néhány könnyen nélkülözhető „kiterjesztett mód" karakter, ezek helyére a matematikai szövegeknél szükséges, ugyancsak szoftverből változtatható görög betű került. A nyomtató és a TASWORD 2 adottságait felhasználva, a szövegszerkesztő vezérlőkaraktereivel például index és hatványkitevő is ábrázolható.

Meggyőződésem, hogy a Spectrum és a TASWORD 2 az irodai és az otthon végzett, például fordítói és szakírói munka nagyon olcsó és rendkívül hatékony eszköze lehet. Ezért igen üdvösnek tartanám, ha lapjukban egy TASWORD termékismertető megjelenne. Magamnál érdemesebb jelentkező hiányában megírását szívesen el is vállalnám.

Véleményem szerint — sokan mások is mondják — a következő néhány évben elsősorban a szövegszerkesztő rendszerek fognak nagy számban elterjedni. Ez a rendszer sikeres lehet, hiszen olyan sok Sinclair van az országban, amelyeket „okos írógépként" részben az ékezetes betűk, részben a nyomtató hiánya miatt nem használnak. A TASWORD 2 lehet a megoldás egyik része, ezért várjuk a cikkét.



Szegedi Gyula grafikái (lásd még 15. és 29.old.)
(EMG–777 típusú gép, robotron 1154–0454 típusú nyomtató)

Makula Tibor 8. oszt. tan. Orosháza,

Tass u. 1. 5900

Megkaptam a kit-gépről szóló levelet. A nyárom szüleimtől kaptam egy C16-os gépet, így a kit-gépre már nincs szükségem. A gépből Magyarországon még sajnos kevés van, így nehéz róla leírást és hozzá programokat szerezni.

Legyen szíves megírni, hogy az NJSZT-nek a Commodore szekciói között van-e C16-tal foglalkozó, és hol lehet erről a gépről részletes leírást kapni (memóriatérkép stb.), esetleg programokat vásárolni, cserélni.

Sok sikert kívánok a C16-hoz. Simonyi Endre a kérdésekre a következő választ adta: A magyar nyelvű irodalmat a Novotrade Rt.-nél (Bp. XIII., Fürst Sándor u. 24—26.) lehet megvásárolni. A HCC Commodore szekciója (Budapest) foglalkozik C16-os gépekkel is.

Kovács János, Tiszavasvári,

Krúdy Gyula u. 6.

Lapjukat tüzetesebben áttanulmányozva feltűnt, hogy a HT-gépre közölt rövid programok tulajdonképpen képrajzolások, amelyek dekoratív megjelenésükkel hatnak. Eszembe jutott, hogy nekem is van olyan programom, amely hasonló feladatot old meg, de bizonyos gyakorlati haszna is lehet.

Néhány hónapig birtokomban volt egy VC—20-as számítógép. Annak a gépkönyvében találtam egy rövid mintaprogramot, amely a CHR$ felhasználásával egy labirintushoz hasonló ábrát rajzolt a képernyőre. A program a következő volt:

```
1 PRINT "CLR HOME"
2 PRINT CHR$(201.5+RND(1));
3 GOTO 2
```

Ez nem elégített ki, mivel csupán két karaktert használt fel, és változtatni csak új programsor írásával lehetett. Az ötlet azonban nem hagyott nyugodni, és kidolgoztam az alábbi rövid, de csodálatos hatású kis programot:

```
10 LET A=INT(RND(1)*255)+1
20 POKE 36879,A
30 PRINT "CLR HOME"
40 LET B=INT(RND(1)*101)+90
50 FOR T=1 TO 440
60 PRINT CHR$(B+0.5+RND(1));
70 NEXT T
80 GOTO 10
90 REM MINTA RAJZOLAS, KOVACS TV
```

A program futtatása során a legváltozatosabb minták és színek jelentkeznek a képernyőn, kb. 10 másodpercenként váltva egymást. Esetenként előfordulhat, hogy a képernyő üres lesz, de ez nem jelenti a program leállását, csak azt, hogy a 101, illetve 90-től felhasznált karakterkódszámok között üres is van, és ilyenkor nem tud a gép miből rajzolni. A minták között számtalan olyan is akad, amely egy iparművészt akár ruhaanyag- vagy tapétaminta alkotására ösztönözheti.

Ha egy minta megtetszik, a RUN STOP gomb megnyomásával rögzíthető a képernyőn, ahonnan lefényképezhető. De lehetőség van arra is, hogy a PRINT A, B közvetlen paranccsal a jellemző két adatot a géptől megkapjuk. Ezt feljegyezve, a program 10-es és 40-es sorát elhagyva, a meglevő A és B értéket közvetlenül beírva, a mintát reprodukálhatjuk.

A CONT parancs beírásával a mintakészítést tovább folytatja a gép. Természetesen a minták igazán csak színes televízión mutatnak.

Gyakorlatok késleltetőkkel

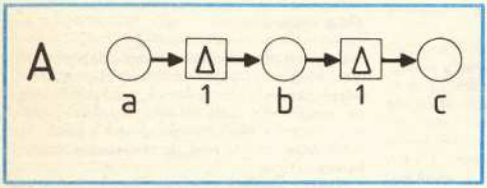
Technikai szünet után — kis kitérőként — közönséges késleltetőkkel kapcsolatos egyszerű feladatokat adunk. A megkezdett utat természetesen ott folytatjuk majd, ahol abbahagytuk.

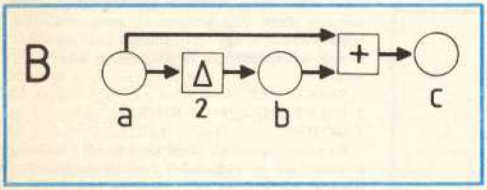
A feladat: meg kell állapítani, hogy a (számmal jelölt) működéstörténeteket mely (betűvel jelölt) rendszer megfigyelésével vettük fel (regisztráltuk).

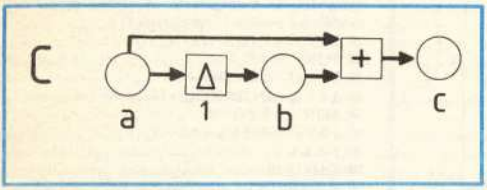
A koordinátarendszer egységeit nem jelöltük, azok ugyanis jól láthatók az ábrákon. A kezdőpont mindig a 0 és 0 koordinátájú pontban van. Az időtengely „fentről-lefelé" állású.

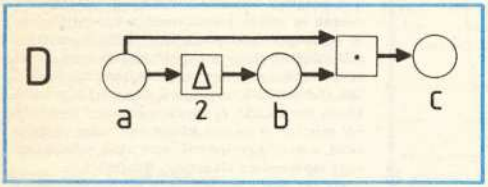
A vizsgálat kezdetén (a 0 időpontban) az összes feladat összes késleltetőjének tartalma az azonosan zérus függvény megfelelő hosszúságú darabkájával írható le, másszóval az összes késleltető 0-val van feltöltve.

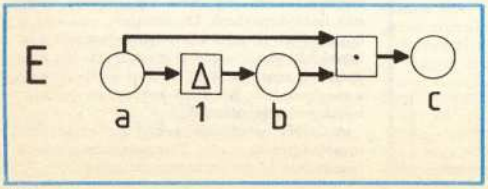
TAKÁCSY ILDIKÓ

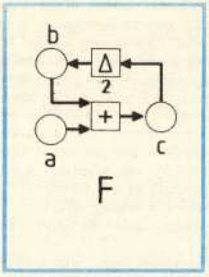


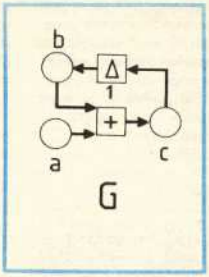


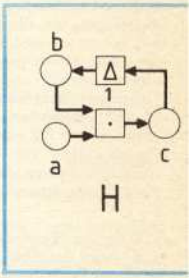


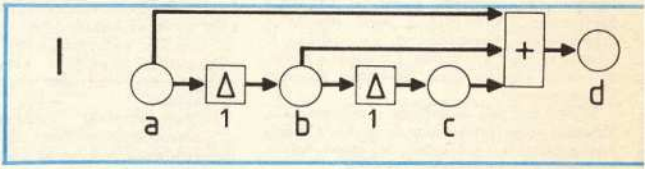


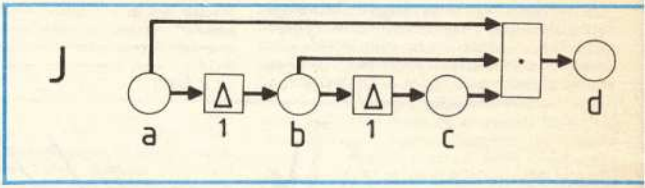


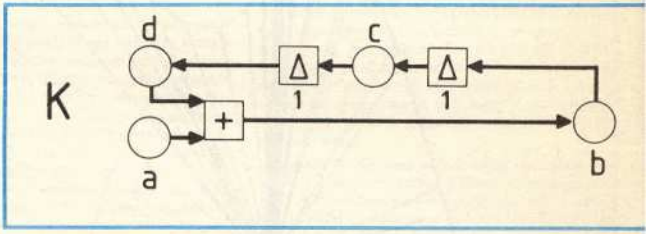


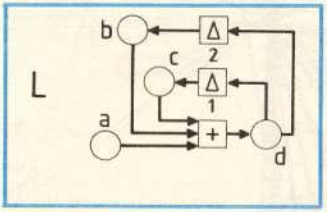


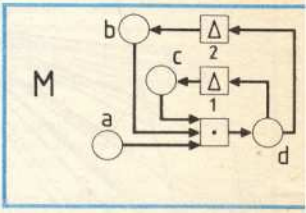


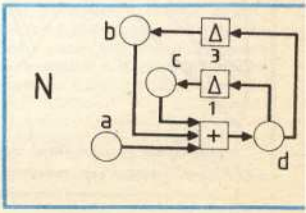


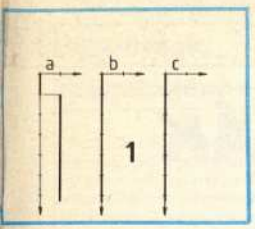


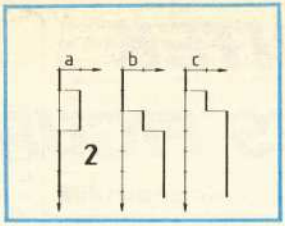


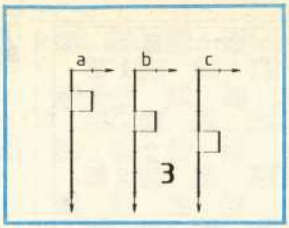


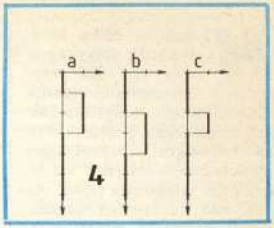


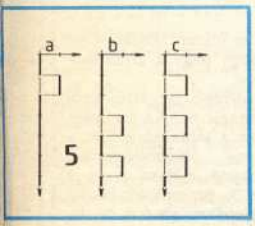


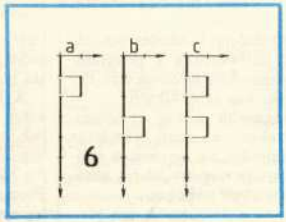


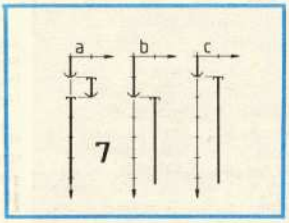


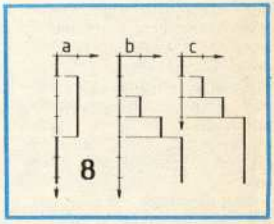


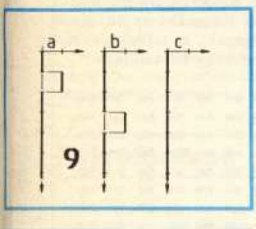


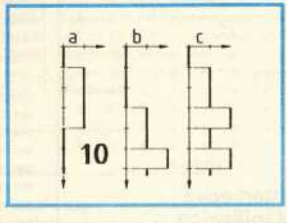


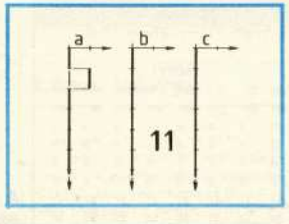


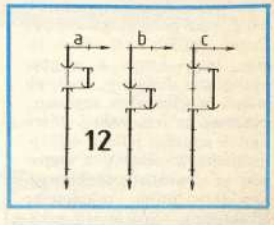


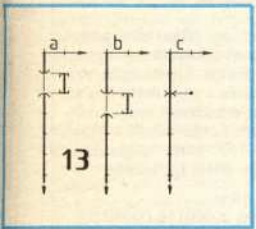


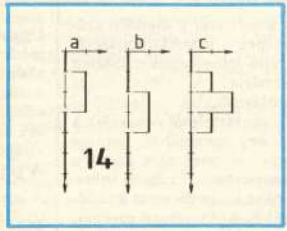


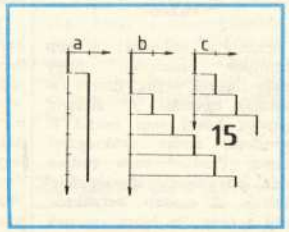


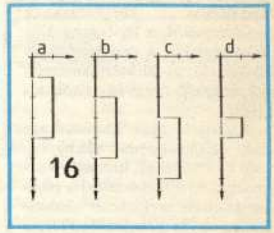


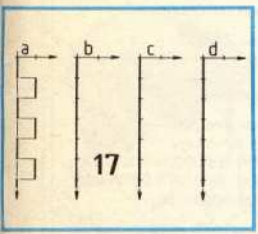


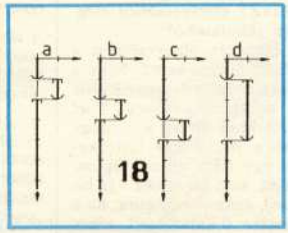


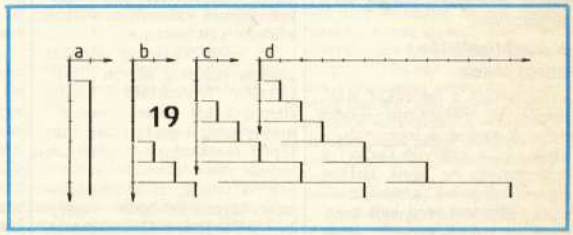


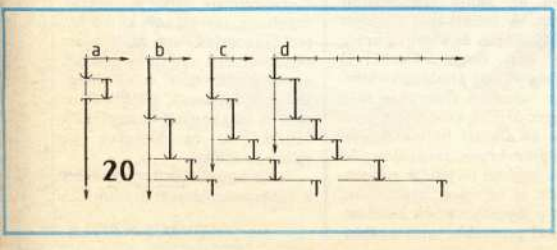


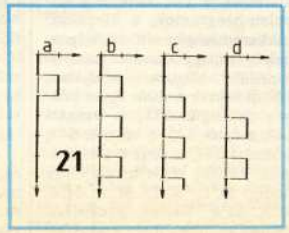












A FELADAT MEGOLDÁSA

1–H, 2–G, 3–A, 4–E, 5–F, 6–B, 7–G, 8–G, 9–D, 10–F, 11–H, 12–C, 13–E, 14–C, 15–G, 16–J, 17–M, 18–L, 19–N, 20–L, 21–K;

illetve A–3, B–6, C–12, C–14, D–9, E–4, E–13, F–5, F–10, G–2, G–7, G–8, G–15, H–1, H–11, I–18, J–16, K–21, L–20, M–17, N–19.

BITEK ÉS FIGURÁK

Hazánkban eddig kevesen írtak sakkprogramot. Az 1985 szeptemberi amszterdami ötödik sakkszámítógép-világbajnokságot megelőzően Hollandiában az országos bajnokságon nyolcvanan vettek részt. Az érdeklődés világszerte óriási. Az 1983-ban Budapesten megrendezett harmadik világbajnokság — amelynek még nem volt magyar résztvevője — nálunk is erősen fokozta az érdeklődést. A Sakkszövetség Számítógép Bizottságánál egyre többen jelentkeznek segítségért, útmutatásért, közöttük egyetemi hallgatók, mérnökök, programozó matematikusok, orvosok.

Vajon mi az oka annak, hogy nálunk a nagy érdeklődés ellenére olyan kevesen vállalkoznak programkészítésre? Ennek legfőbb okát abban látom, hogy kevés a magyar nyelvű szakirodalom, amely elméleti és gyakorlati segítséget nyújtana az induláshoz. Mérő László elméleti jellegű cikksorozatához kapcsolódva szeretném az olvasót a konkrét programozástechnika rejtelmeibe elkalauzolni. Szó lesz a sakkhadállások ábrázolásának rendszereiről, a lépésgenerálás módszereiről, a program tagozódásáról, az állásértékelésről, a kiértékelő függvény funkciójáról.

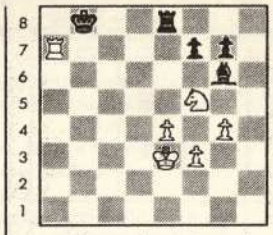
Azoknak, akik kedvet kapnak számítógépes sakkprogram készítéséhez, bizonyára segítséget nyújtanak majd a cikksorozatban megjelenő szubrutinok, egyes programrészletek. Előbb azonban ismerjük meg az alapokat!

A sakkhadállás ábrázolása

Az ember a figurákat alakjukról és méretükről ismeri meg. A gyalog a legkisebb, a király és a királynő (vezér) a legnagyobb. Az egyik játékos figurái világosak, a másiké sötétek. Mindezt meg kell mondani a számítógépnek, hogy megértse, hol vannak a figurák a táblán, mert csak így tudja kiszámítani a bábuk lehetséges lépéseit. Ezt úgy lehet megoldani, hogy a különféle típusú figurákat eltérő számokkal jelöljük.

A táblaindexes módszer

Assembler nyelvű programokban általában a következő rendszert alkalmazzák. A számok hexadecimális (16-os számrendszerbeli) adatként értendők: a világos gyalog $1, a huszár $2, a futó $3, a bástya $4, a vezér $5, a király $6; sötét megfelelőjük pedig az $81, $82, $83, $84, $85, $86 számot kapja. Az üres mezőket $0-val jelöljük. Ennek alapján a sakkjátszma kiinduló állása a következőképpen ábrázolható:



SÖTÉT							
$84	$82	$83	$85	$86	$83	$82	$84
$81	$81	$81	$81	$81	$81	$81	$81
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
$1	$1	$1	$1	$1	$1	$1	$1
$4	$2	$3	$5	$6	$3	$2	$4

VILÁGOS

Ezzel a jelöléssel a gép könnyen „észreveszi", hogy mely figurák védik, illetve támadják egymást. Az állásról viszont közel sem mond el mindent, amire szükségünk lenne. Például nem tudjuk meg, hogy mely figurák mozdultak el eredeti helyükről, ami a sáncolás lehetőségének egyik alapfeltétele. Ezt a számítógépnek valamilyen módon számon kell tartania.

Az egyik lehetőség mindkét játékos számára három „flag" (jelzőbit) fenntartása a király, illetve a két bástya számára, amely jelzi, hogy történt-e már lépés ezekkel a bábokkal, vagyis van-e sáncolási lehetőség. A másik, az általánosabban elterjedt módszer — melylyel a Sargon, a Grandmaster, a Caissa, a The Turk, a Kempelen-programok, a Mephisto sakkszámítógép stb. is dolgoznak —, hogy minden már elmozdult figura kódjához $40-et adnak hozzá. Így a világos gyalogot $41, a huszárt $42, a futót $43, a bástyát $44, a vezért $45, a királyt $46 számok jelölik; sötét megfelelőik pedig $C1, $C2, $C3, $C4, $C5, $C6. Ennek gyakorlati előnye az, hogy ilyen kódok alkalmazásánál feltételes utasításokat iktathatunk a programba. Assembler nyelven erre külön utasítás is található.

Magasabb szintű programnyelveken a fenti rendszer megfelelője az, amikor a világos figura negatív kódja ábrázolja a sötét bábokat.

SÖTÉT							
−4	−2	−3	−5	−6	−3	−2	−4
−1	−1	−1	−1	−1	−1	−1	−1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
4	2	3	5	6	3	2	4

VILÁGOS

A bittérképes ábrázolásmód

Ez a módszer a digitális számítógépek utasításrendszerében rejlő lehetőségeket jobban kihasználja. A táblaindexes ábrázolásmóddal ellentétben nem a tár tartalmát rendeljük a tábla egy mezőjéhez, hanem egy-egy 64 bites szót minden figuratípushoz. Értékes információkat kaptunk erről a módszerről T. A. Marsland professzornak 1984 augusztusában az NJSZT szervezésében megtartott előadásából.

A bittérkép elnevezést ez a módszer onnan kapta, hogy a 64 bites szó minden egyes bitje egy sakktáblamezőnek felel meg. A bitek értéke jelzi, hogy abban a pozícióban tartózkodik-e egy adott típusú figura. Például, ha a bit értéke 1 (a bit magas), akkor van figura, ha a bit értéke 0 (alacsony), nincs figura az általa reprezentált helyen. A bittérképet minden típusú világos és sötét figurára létre kell hozni. Ezenkívül szükség van az általános bittérképre, amely a figurákra való tekintet nélkül, külön jelzi a világos és a sötét bábuk helyét.

A bittérképes ábrázolásmód egyre jobban terjed. A programozók is szívesen használják, mert a lépésvariációk kiszámításánál gyorsabb, mint a táblaindexes módszer. Hátránya viszont, hogy nagy a tárkapacitás-igénye.

A könnyebb megértés kedvéért konkrét példán hasonlítjuk össze a táblaindexes és a bittérképes ábrázolásmódot. A Mephisto A'dam 2 és a Plymate X programok között az amszterdami vb-n a 39. lépés után az *ábrán* látható hadállás jött létre. Ezt az állást a számítógépek a táblaindexes módszerrel így ábrázolják:

$00	$C6	$00	$00	$C4	$00	$00	$00
$44	$00	$00	$00	$00	$81	$81	$00
$00	$00	$00	$00	$00	$00	$C3	$00
$00	$00	$00	$00	$00	$42	$00	$00
$00	$00	$00	$00	$41	$00	$41	$00
$00	$00	$00	$00	$46	$41	$00	$00
$00	$00	$00	$00	$00	$00	$00	$00
$00	$00	$00	$00	$00	$00	$00	$00

Ugyanez az állás bittérképes módszerrel a következőképpen alakul. Külön bittérképük van a világos és a sötét bábuk által elfoglalt mezőknek, majd szintén külön-külön minden egyes világos és sötét figurának. A gép az állást így tárolja:

A sötét állás
01001000 00000110 00000010
00000000 00000000 00000000
00000000 00000000

A világos állás
00000000 10000000 00000000
00000100 00001010 00001100
00000000 00000000

A világos gyalogok szerint az állás bittérképe
00000000 00000000 00000000
00000000 00001010 00000000
00000100 00000000

A könnyebb áttekinthetőség kedvéért a 64 mezőt soronként tagoltam; így alakult ki a 8 bites tagozódás, ami a 8 bites társzervezésű számítógépeknél egy-egy bájtnak felel meg. Eszerint folytatva, a többi figurára is megkapjuk a megfelelő bittérképet, és kialakul az egész hadállásé.

Ismertetésünket legközelebb a lépésgenerálással folytatjuk.

KOVÁCS ATTILA

Rajzfilmek számítógéppel

A rajzfilmek rendkívül munkaigényesek, hisz a rajzolóknak minden egyes műsormásodperchez 24 önálló képet kell megrajzolniuk és kiszínezniük. A hagyományos eljárás során ezeket a fázisrajzokat egymás után lefényképezik, összevágják, és ezzel kész is a film.

A számítógép segítségével már kísérleteket folytatnak olcsóbb filmek előállítására. A rajzoló fényceruzával a képernyőre rajzolja a kívánt alakot. Ezt a képet a gép pontokra bontja. Az angol—japán együttműködésben készült animációs technika lehetővé teszi, hogy a rajzoló számítógépprogram segítségével tetszőlegesen mozgassa a képernyőre rajzolt figurát. A gép az egyes mozgásfázisokat háttértárolójára kiírja.

Jelenleg azonban nincs olyan számítógép, amely az elraktározott adattömegből a képeket a szükséges 24 kocka/másodperc sebességgel tudná a képernyőn megjeleníteni, ezért most minden képet egymás után, önállóan kell kimerevíteni a képernyőn, és a készülék elé állított filmkamerával sorban felvenni az egyes kockákat.

Mikrogépek Távol-Keleten

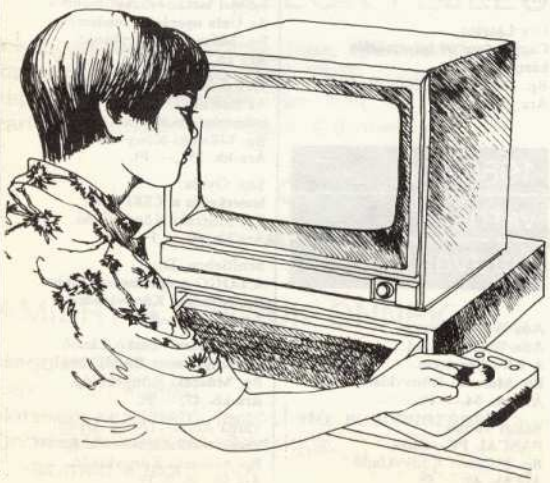
Dél-Koreában a legelterjedtebb személyi számítógépek jelenleg az Apple II-nek ott készülő funkcionális megfelelői. Nagyobb teljesítményt igénylő feladatoknál az IBM PC-t használják.

Japánban viszont az Apple II egyáltalán nem terjedt el, mivel ára viszonylag magas. Az IBM PC is még ritkán fordul elő, mert még nem alkalmas a japán szövegfeldolgozásra. Helyette a japán IBM-nek az IBM 5550 jelzésű gépét forgalmazzák nagy tömegben, amit japán szövegfeldolgozó gépként hirdetnek.

Rajzológömb

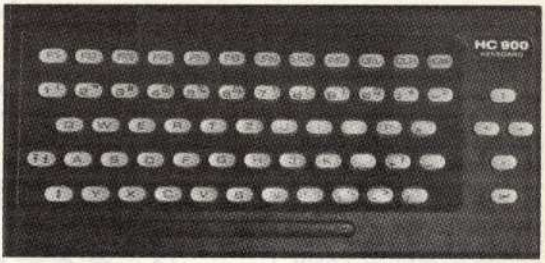
A jó repülőgépműszereiről ismert angol Marconi cég rajzológömb (trackerball) elnevezéssel érdekes kis eszközt hozott forgalomba mikroszámítógépeken való felhasználásra. Egy műanyagdobozból kb. egyharmadrésznyire kiálló, egyetlen ujjal könnyedén minden irányban mozgatható, 5,7 cm átmérőjű fémgömb egy fordulatra mind az x, mind az y tengely irányában 200 jelet küld ki a hozzákapcsolt számítógépbe. Tehát, 1,8 mm-enként vagy 1,8 fokonként 1-1 jelet.

Az eszköz a képernyőn való szabadkézi rajzoláshoz, festéshez kiválóan alkalmas. Használható műszaki rajzok készítésére, így a számítógéppel segített tervezésnek is kiváló segédeszköze lehet. A játékokhoz használt botkormányt és az újabb számítógépeken a kurzor mozgatására szolgáló „egeret" is jól helyettesíti. A dobozon van még 3, szabadon programozható hatású nyomógomb is. Egyelőre az angol iskolaszámítógéphez vagy a Commodore 64-hez csatlakoztathatóan szállítják. Bolti ára 60 angol font.



NDK háziszámítógép

Egyre népszerűbbek az NDK-ban a mühlhauseni Mikroelektronikai Gyárban készülő HC900 típusú háziszámítógépek. Ebben meghatározó szerepet játszanak a gép kiváló tulajdonságai: a 16 szín megjelenítési képessége, a finom, 320 × 256 pontos grafika, valamint az 5 oktávos hanggenerátor. A Z80 mikroprocesszornak az NDK-ban gyártott változatával, az U880-nal működő gép operatív tára 32 kbájt. A felhasználó számára a CAOS operációs rendszer biztosítja az alapfunkciókat. A felhasználói prorgramokat BASIC nyelven lehet írni.



Lézerkártya

Kiszivárgott hírek szerint a japán cégek a közeljövőben megjelennek a piacon olyan lézerkártyákkal, amelyek műszaki paraméterei lényegesen felülmúlják a hagyományos integrált áramköri kártyákét. Árban is versenyképesebbek lesznek, max. egy dollárba fog kerülni darabja. E kártyák teljesítményét és alkalmazási körét illetően még nincs részletesebb információ.

Zuhanó árak

A mikrogépek kínálatának rohamos növekedése miatt az 1985. év az árrobbanás jegyében kerülhet be a hazai számítástechnika-történetbe. Az év elejétől a végéig mintegy 50 százalékkal csökkentek az árak.

A háziszámítógépek kategóriájában a 48 k-s Spectrum ára az 1984. évi 40 ezer forintról 1985 végére már 17—20 ezerre zuhant. Hasonlóképpen felére csökkent ez időszak alatt a Commodore 64 ára is, mely ma már 50 ezer forint alatt kapható.

Az IBM PC ára tavaszig mutatott emelkedő tendenciát, majd rohamosan zuhanni kezdett. Ebben közrejátszott a vele kompatibilis és nála jóval olcsóbb nyugati típusok hazai megjelenése is. A hazai gyártók is rugalmasan reagáltak a változásokra: a magyar gyártmányú, IBM PC-vel kompatibilis gépek ára őszre 400 ezer forint alá csökkent.

Természetesen a fenti árak csak a központi egységekre vonatkoznak, a perifériákkal ellátott rendszerek ára ennél jóval magasabb.

Elembázis- és súlycsoportváltás

Az éveken át homogén, csak a Motorola 6800 mikroprocesszor funkcionális ekvivalensét tartalmazó bolgár, 8 bites mikroszámítógépek gyártásába 1984 elején ékelődött be az első „rendszeridegen" elem: az IZOT 1031 már Z80-as mikroprocesszort tartalmazott, és a CP/M operációs rendszer vezérlése alatt működött. Ez egy olyan lavinát indított el, mely másfél év múlva, őszre érett be. Minden átmenet nélkül, egy csapásra eltűntek a régi bázisú gépek, és ugyanilyen hirtelen szűnt meg a rájuk folyó szoftverfejlesztés is. A Motorola 6800 alapú gépek tulajdonosainak az átállás megkönnyítésére megjelent az ORGTECH 80/600-as család.

Ezek kétprocesszoros gépek, és a nevükben levő két szám mutatja az eredetüket. A 80 a Z80-as, a 600 pedig az SZM 600-as (ezen a néven gyártják Bulgáriában a Motorola 6800 funkcionális ekvivalensét) mikroprocesszorra utal. Ezek a gépek a Motorola MDOS operációs rendszere mellett már a világon jóval elterjedtebb CP/M operációs rendszert is ismerik.

Az ORGTECH 80/600-as család méreteit tekintve is újdonság. Asztalra tehető, billentyűzetbe épített, elegáns gépet készítettek.

MŰSZAKI KÖNYVÁRUHÁZ

1071 Budapest VI., Liszt Ferenc tér 9.
Levélcím: 1414 Budapest, Pf. 79.

Szakkönyv-kínálat

Appel György—Mihályfi János:
A HT—1080Z iskolaszámítógép programozása.
Bp. 1985. Interpress,
147 oldal. Ára: 66,— Ft.

Appel György—Kőhegyi János—Zsakó László:
Számítógépes feladatok
(Példatár
személyi számítógépekre).
Bp. 1985. Interpress,
131 oldal. Ára: 66,— Ft.

D. H. Beil:
Adatállomány-feldolgozás COBOL programnyelven.
Bp. 1985. Műszaki Könyvkiadó,
303 oldal. Ára: 82,— Ft.

Bodor Tibor—Gerő Péter:
A Commodore 64 programozásának gyakorlata.
Alapismeretek.
Bp. 1985. SZÁMALK, 160 oldal.
Ára: 55,— Ft.

Bradbeer, R.—DeBono, P.—Laurie, P.:
Műsoron a számítógép.
Bp. 1984. Műszaki Könyvkiadó,
211 oldal. Ára: 90,— Ft.

Brückner Huba:
Videotex-rendszerek és alkalmazásaik.
Bp. 1985. SZÁMALK, 277 oldal.
Ára: 136,— Ft.

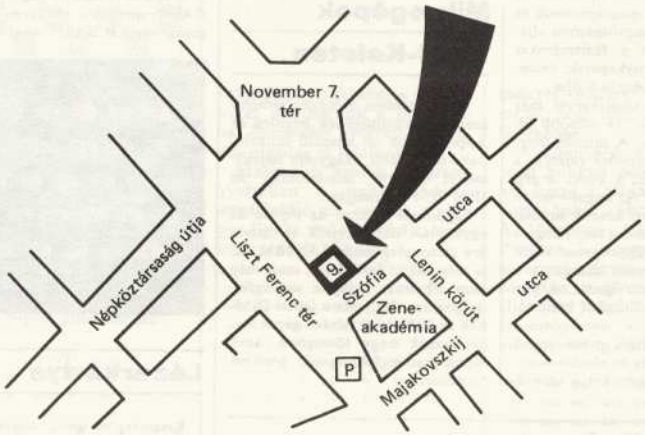
J. W. Coffron:
Mikroprocesszoros rendszerek gyakorlati hibakeresése.
Bp. 1984. Műszaki Könyvkiadó,
244 oldal. Ára: 80,— Ft.

Csákány Antal—Vajda Ferenc:
Játékok számítógéppel.
2. kiad.
Bp. 1985. Műszaki Könyvkiadó,
284 oldal. Ára: 57,— Ft.

Gálfi Zoltán—Ivanyos Lajosné:
Macro nyelvű programozás.
1—2. kötet.
Bp. 1984. SZÁMALK, 580 oldal.
Ára: 240,— Ft.

Kőhegyi János:
Ismerd meg a BASIC nyelvjárásait! HT—1080Z; ABC—80; ZX81.
Bp. 1984. Műszaki Könyvkiadó,
164 oldal. Ára: 59,— Ft.

Kőhegyi János:
Ismerd meg a BASIC nyelvjárásait! ZX—Spectrum; TI—99/4 A; Proper—16/A.
Bp. 1985. Műszaki Könyvkiadó,
185 oldal. Ára: 65,— Ft.



Lőcs Gyula—Vigassy József:
A FORTRAN programozási nyelv.
6. bőv. kiad.
Bp. 1985. Műszaki Könyvkiadó,
433 oldal. Ára: 77,— Ft.

Marschik Iván:
Mikrogéprendszerek tervezése.
1—2. kötet.
Bp. 1985. SZÁMALK, 766 oldal.
Ára: 439,— Ft.

Sztrókay Kálmán:
A Z80 assembler HT—1080Z számítógépes példákkal.
Bp. 1985. Műszaki Könyvkiadó,
218 oldal, függelékkel.
Ára: 74,— Ft.

Bartha Tamás:
Mikrogépek illesztése. Az IEC-busz és alkalmazása.
Bp. 1985. LSI ATSZ,
379 oldal. Ára: 341,— Ft.

Dibuz Ágoston:
Hardware katalógus. Alkatrészek, elemek.
Bp. 1985. LSI ATSZ,
250 oldal. Ára: 31,— Ft.

Dobay Péter:
Mikroszámítógépes program-katalógus.
Bp. 1985. LSI ATSZ.
Ára: 178,— Ft.

Donát János:
Z80 software-táblázatok.
Bp. 1985. LSI ATSZ,
53 oldal. Ára: 140,— Ft..

Kovács Magda:
Egyszerűen a mikroszámítógépről.
Bp. 1985. LSI ATSZ, 321 oldal.
Ára: 156,— Ft.

Krizsán György:
Zilog mikroprocesszor-családok.
1—2. kötet.
2. jav. bőv. kiad.
Bp. 1984. LSI ATSZ, 452 oldal.
Ára: 318,— Ft.

Salgó Iván:
Hardware-katalógus. Hazai készülékek.
Bp. 1985. LSI ATSZ, 424 oldal.
Ára: 51,— Ft.

I 8086 mikroprocesszor. Utasításkészlet.
Bp. 1985. LSI ATSZ, 259 oldal.
Ára: 239,— Ft.

Úry László:
Commodore 64 információs kártya.
Bp. 1985. LSI ATSZ, 39 oldal.
Ára: 95,— Ft.

1986-ban várható kiadványok

Ada-Winter Péter—Ada-Winter Dávid:
A ZX—Spectrum.
Bp. Műszaki Könyvkiadó.
Ára kb. 54,— Ft.

Bakos Tamás:
PASCAL PC-seknek.
Bp. Műszaki Könyvkiadó.
Ára kb. 49,— Ft.

Csépai János:
A számítástechnika alapjai.
Bp. Műszaki Könyvkiadó.
Ára: kb. 63,— Ft.

Dacosta, F.:
Kalandok BASIC-országban.
Bp. Műszaki Könyvkiadó.
Ára kb. 60,— Ft.

Kőhegyi János:
Ismerd meg a BASIC nyelvjárásait! Commodore 64; VIC 20; Sharp 1500.
Bp. Műszaki Könyvkiadó.
Ára kb. 59,— Ft.

Kőrösi István—Szabó Sándor:
Az Unix operációs rendszer.
Bp. Műszaki Könyvkiadó.
Ára kb. 52,— Ft.

Newman, W. M.—Sproull, R. F.:
Az interaktív számítógépes grafika.
Bp. Műszaki Könyvkiadó.
Ára kb. 157,— Ft.

Sági Gyula:
Ismerkedés a ZX81-gyel.
Bp. Műszaki Könyvkiadó.
Ára kb. 60,— Ft.

Senftleben, D.:
A LOGO programozási nyelv.
Bp. Műszaki Könyvkiadó.
Ára kb. 78,— Ft.

Szlávi Péter—Zsakó László:
Így programozz BASIC-ban!
Bp. Műszaki Könyvkiadó.
Ára kb. 47,— Ft.

Varga Antal—Dusza Árpád:
Programozási ismeretek. BASIC.
Bp. Műszaki Könyvkiadó.
Ára kb. 66,— Ft.

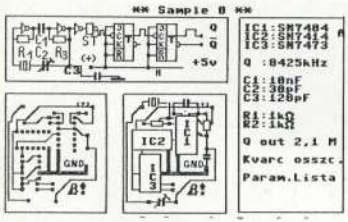
SUPER GRAFIKA C64-RE „ÁGI" nagy felbontású grafikai programcsomag

A programcsomag segítségével megrajzolhatja és saját programjába átmásolhatja mindazokat az Ön által tervezett képeket, rajzokat, melyek egy 320 × 200-as raszterben elférnek.

A grafikai lapon a C64 standard karakterkészletével tetszőleges szöveg helyezhető el. A billentyűzetről bevitt paraméteres alakú függvények grafikai rajzai is megjeleníthetők. A „szabadkézi" rajzolás botkormány segítségével oldható meg.

A félig vagy teljesen elkészített rajzok floppylemezen tárolhatók, így bármikor lemásolhatók az Ön saját programjába. Definiálható a programmal 5 × 3 karaktertömb is, mely képernyőméretűvé nagyítható.

Megrendelhető:



A program illeszthető a KOALA POINT rendszerhez is, így még számos alkalmazási lehetősége van.

A grafikák, rajzok MPS 801 vagy 802 típusú sornyomtatón is megjeleníthetők.

ALKOTÓ IFJÚSÁG EGYESÜLÉS

Alkotó Ifjúság Egyesülés
Számítástechnikai Iroda
Budapest V., Garibaldi u. 2
Levélcím: 1519 Budapest, Pf.: 330
Telefon: 112-666, 113-608
Telex: 22-7272

Lehet, hogy Ön az elektronika specialistája..., de van, amihez mi értünk!

FÓLIAKLAVIATÚRÁK, TASZTATÚRÁK, MEMBRÁNKAPCSOLÓK FEJLESZTÉSE ÉS GYÁRTÁSA

a kapcsolóelemek száma: max. 150 nyomógomb
kapcsolási feszültség: max. 24 V
kapcsolási teljesítmény: max. 1 W
zárt kontaktusok ellenállása: 0,5 ohm

kivitel:
pornak, víznek, vegyszereknek és egyéb szennyeződéseknek ellenálló, lemosható

GRAFIKAI KIALAKÍTÁS ÉS MÉRET A FELHASZNÁLÓ IGÉNYE SZERINT!

AMIÉRT AJÁNLJUK ÖNNEK:

Megbízható
Lapos, zárt kivitelű
Könnyen kezelhető, lemosható, szennyezésálló
Ön megálmodja, mi elkészítjük . . .
. . . és mert a legolcsóbb!

KISIPARI TERMELTETŐ VÁLLALAT

Gyártja és forgalmazza
a Kisipari Termeltető Vállalat
Bp. V., Tanács krt. 24. 176-822
A gyártó egység címe:
Bp. IX., Mátyás u. 5/b 189-175

VT 16 személyi számítógép

16 és 8 bites feldolgozás,
MS/DOS, CP/M 86 és CP/M 3,0 kompatibilis operációs rendszerek,
szoftver-kompatibilitás IBM személyi számítógéppel,
kiváló minőségű grafikus ábrázolás,
távadat-feldolgozás.

VIDEOTON
ELEKTRONIKAI VÁLLALAT

Információ:
VIDEOTON Számítástechnikai Gyára
1021 Budapest, Vörös Hadsereg útja 54.
Telefon: 213-187